AD-A044 234 HUMAN FACTORS RESEARCH INC GOLETA CALIF F/G 5/8
FUNCTIONAL REQUIREMENTS AND OTHER DESIGN FEATURES OF A MANNED S--ETC(U)
AUG 77 G V BAILEY, R T HENNESSY, C D WYLIE N00123-76-C-1245

UNCLASSIFIED NL

1 OF 3
AD
A044234

OF 3

4234

ADA044234

TECHNICAL REPORT 1783-F
CONTRACT NO. N00123-76-C-1245

FUNCTIONAL REQUIREMENTS AND OTHER DESIGN FEATURES OF A MANNED SYSTEM RESEARCH FACILITY

FINAL REPORT

AUGUST 1977

PREPARED FOR
NAVY PERSONNEL RESEARCH AND DEVELOPMENT CENTER
DEPARTMENT OF THE NAVY
SAN DIEGO, CALIFORNIA 92152

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

HUMAN FACTORS RESEARCH, INCORPORATED
SANTA BARBARA RESEARCH PARK, GOLETA, CALIFORNIA

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

2

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	(6) FUNCTIONAL REQUIREMENTS AND OTHER DESIGN FEATURES OF A MANNED SYSTEM RESEARCH FACILITY
5. TYPE OF REPORT & PERIOD COVERED	(9) Final Report, October 1976 – June 1977
6. PERFORMING ORG. REPORT NUMBER	(14) 1783-F
7. AUTHOR(s)	(10) Gregory V. Bailey Robert T. Hennessy C. Dennis Wylie
8. CONTRACT OR GRANT NUMBER(s)	(15) N00123-76-C-1245
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Human Factors Research, Incorporated 6780 Cortona Drive Goleta, California 93017
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	Navy Personnel Research & Development Center Department of the Navy San Diego, California 92152
12. REPORT DATE	(11) August 1977
13. NUMBER OF PAGES	196
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	N/A (12) 194 p.
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

N/A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Manned systems research	Distributed processing
Man-machine interface	Computer programming languages
Human factors research	Operating systems
Display technology	FORTH
Command & control system simulation	Team performance research

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A study was conducted to describe the functional requirements for the development of a Manned System Research Facility (MSRF) to be located at NPRDC. The MSRF will be a laboratory for the demonstration of and experimentation on advanced concepts of man-machine interaction and the controlled study of operator performance in a system context. The report consists of a section providing an introduction to the MSRF concept, and sections dealing specifically with the recommended MSRF subject work station consoles, computer system, software requirements, voice communications system, (over)

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

402 185

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

physical facility, staffing, and acquisition and development plan. The report contains reviews and/or discussions of manned systems research, display technology, distributed processing systems, computer operating systems, and programming languages.

ACKNOWLEDGEMENTS

The authors are indebted to Dr. Frederick A. Muckler, Program Director, Design of Manned Systems, NPRDC, and Dr. Samuel E. Bowser, Personnel Research Psychologist, NPRDC, for their support and patience during the execution of this study.

We also wish to thank the many members of the HFR staff who participated in the arduous tasks of gathering detailed data, conducting experimental evaluations of design concepts, and in physically producing this report.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

List of Figures

LIST OF FIGURES (CONTINUED)

LIST OF TABLES

INTRODUCTION

Purpose

The purpose of this study is to describe the functional requirements for the development of a Manned System Research Facility (MSRF). The MSRF will be a laboratory for the demonstration of and experimentation on advanced concepts of man-machine interaction and the controlled study of operator performance in a system context.

The MSRF will be used to test new man-machine interaction concepts resulting in general principles for the development of new Naval systems. Data will be gathered on both individual and team performance to understand their capabilities and limitations which can consequently be used to support design decisions. The emphasis of the work conducted in the MSRF, therefore, will be on immediately usable products of value to the Navy in current designs for near future Naval systems. Although the MSRF is not limited to use for research on equipment-related variables, it is particularly concerned with the man-machine relationship.

This report is the recommended design for the MSRF including the subject work station consoles, the computer control system, the software system requirements, the physical layout of the facility, the staffing of the facility, and the acquisition and development plan.

General Functional Requirements

The principal functional requirement of the MSRF is to provide for the experimental control and acquisition of data about the flow of information across the man-machine interface for a variety of Navy systems. Therefore, it must be possible to simulate the system's operational characteristics apparent to the operators of the system.

Information presented or displayed to operators of Navy systems can take many forms. Discrete indicator gauges, alphabetic and numeric readouts, and CRT displays with sensor graphic and alphanumeric information are probably the most common means of displays. Because of the extreme sophistication of the displays in some Navy systems it is not practical to expect the MSRF to have the capability of exact faithful reproduction of the displays used in a number of systems. For research purposes, however, the most important feature of the display, the informational content, must be faithfully reproduced in the MSRF. It is difficult to define exactly what is meant by informational fidelity, but, in general, this term is taken to mean that the general appearance of the simulated display is similar to that used in the actual system and that the operator can extract the same information from the display using the same perceptual and/or cognitive processes in approximately the same amount of time that would be required in the actual system.

In addition to acquiring information from a simulated system display, the operator(s) must be able to enter control inputs in a simulated system in a manner similar to that used in the actual system. The MSRF, therefore, must provide input devices similar to those used in existing or near future Navy systems. Typical input devices, such as joysticks and trackballs for cursor control, buttons, switches, and knobs for discrete and variable inputs are relatively easy to duplicate in a simulated system.

Voice communications via sound-powered phone systems, intercoms, ennunciators, radio links, and conventional dial or push-button telephones are used singly or in combination in many Navy systems. The MSRF, therefore, must also have the same number and type of voice communication channels available.

Generally the three types of functions just discussed, display of information, acceptance of control inputs, and

means for voice communications, are integrated into a common unit known as a work station console. Since the function and arrangement of work station consoles are different for different Navy systems, the MSRF must have reconfigurable work station consoles which can be made to appear and function similar to those work station consoles used in a variety of Navy systems.

Because the MSRF will be used for research on man-system interactions, it is necessary not only to simulate the essential characteristics of systems as they exist now, but also to modify and change the functions of the system and the display control and communication devices on the work station console. In the case of research involving a team rather than an individual, it may be necessary to allocate information differently among the consoles or provide entirely different types of information to the console. In other words, a great deal of flexibility is required in both function and appearance of the work station console. Additional flexibility is also required in the type of data that is acquired during the course of the experiment. Different experiments with very different purposes will require different measures of performance. Consequently, the means of specifying the data to be collected and under what conditions must be easily changeable.

It is anticipated that research in the MSRF can involve up to eight subjects working at three separate consoles. Each console may have one or two CRT display units. Therefore, the computer system must be able to support a minimum of six primary CRT displays plus a variety of control devices required for three consoles. The physical space available must be sufficient to comfortably house three fairly large consoles and eight people. It is also assumed that the consoles can be arranged within the room in several configurations.

It is anticipated that the MSRF will have staff programmers to do the systems and application programming. It is also likely, however, that experimenters themselves or

personnel who work for them will also participate in the preparation of applications programming. Consequently, the MSRF programming system should be relatively easy to use and not require a great deal of programming sophistication or require extensive periods of time to prepare research applications programming.

A last general requirement for the MSRF is that it be easily expandable, if necessary, to meet future needs. It is expected that the MSRF will be a viable test bed for research on man-machine interaction in Navy systems throughout the 1980s. Since it is impossible to anticipate all future systems requirements and research related thereto, the MSRF, and particularly the computer system, must allow for expansions to meet future research needs.

Research Capability Requirements

Before beginning work on the detailed specification of the MSRF facility and equipment requirements, a survey was conducted of all NPRDC departments which were likely to use the MSRF. The following questions were included in the survey.

1. For what type of research would you use the MSRF? If appropriate, mention the current or anticipated Navy systems to which the research is relevant.

2. To what extent is simulation display fidelity important to your research needs? That is, is it necessary that displays accurately simulate actual operational displays in content, format, and dynamic characteristics?

3. It can be assumed that automatic performance monitoring will be part of the MSRF. That is, all subject inputs to the equipment controls can be recorded automatically. Do you have any special or additional data acquisition needs which do not come from subject inputs, such as physiological measures, observational data, video or audio recording, etc.?

4. A major characteristic of a computer-controlled laboratory is that the stimulus sequence of simulated situation scenario can be programmed to be contingent upon the actions of the subject(s). It is also possible to allow the experimenter to intervene or control the experiment manually. If you envision the need for experimenter intervention in your work, please describe under what general circumstances and in what way you would wish the experimenter to manually intervene.

5. The initial staffing of the MSRF would probably consist of *at least* a laboratory director and a knowledgeable system and application programmer. Would you prefer to have any programming necessary for your research to be done by the MSRF staff, or would you prefer to use your own personnel for the programming of the experiment?

6. Are you currently using any equipment which you would want to use in the MSRF and would not have to be part of the resident MSRF equipment?

7. If you will study team performance, what is the maximum number of team members which would make up the experimental group?

8. It may be possible to place the laboratory in a mobile van or trailer. If applicable, please describe briefly how mobility would be advantageous for your research.

9. Do you expect to use classified material in your research? If so, would it be in hard copy or computer-storable form?

10. Please add any additional comments which will help to clarify what features or equipment you would like to have in the MSRF.

The results of the survey are shown in Table 1. The first column shows the departments surveyed by code and title and the second column shows the basic type of research the department is concerned with. The other columns are topically keyed to the questions asked in the survey. The response by each code is severely abbreviated in each column.

TABLE 1
SURVEY RESULTS

CODE	TYPE OF RESEARCH	NAVY SYSTEM	DISPLAY FIDELITY	SPECIAL EQUIPMENT REQUIREMENTS	EXPERIMENTER INTERVENTION REQUIREMENTS	PROGRAMMING RESPONSIBILITY	OWN EQUIPMENT TO USE IN MSRF	MAXIMUM TEAM SIZE	MOBILITY DESIRED, REASON	CLASSIFIED MATERIAL USE IN MSRF	COMMENTS
301 - Management of People and Organizations	Organizational Structure Effects on Performance	Automatic propulsion systems Manned computerized systems	Functional Fidelity	Audio/Video Recording	None	MSRF	None	6-20-40	Yes, Subject Accessibility	Possible	Data Transfer to Other Computers Needed
302 - Measurement of Job Performance	Job Performance	Sonar Test Equipment Propulsion Systems	Important (Effects of Fidelity on Sonar Performance)	Video Tape	Yes	MSRF	None	6-7	Yes, Subject Accessibility	Yes - Sonar	
303 - Management Systems	Manager-Model Interface Problems (MSRF Probably of Little Use)		Very Little Importance	Large Scale Random Access Files and Model Inputs, No Other	Not Sure	Own Programming	Yes - Unspecified	1	No	No	
304 - Development of Training Technology	Operator and Tactical Team Training	NTDS, ASW, CIC	Fidelity Primary	EEG, EMG, EKG, Audio/Visual	Stop Action and Replay	Both	No	6-8	Yes, Subject Accessibility	Some Possibly	
305 - Information and Decision Processes	C^3 and Management Aiding Information Systems	NTDS Mini-NAMPS	Informational and Symbolic Fidelity Important	None	Restart with Different Parameters	Own Programming	None	Few	Yes, Subject Accessibility	Yes	
306 - Test and Application of Training Systems	Symbology Types of Maps	A Marine Tactical Command and Control System	Content and Symbology Important	None	Temporary Stop for Instructions	MSRF	None	3	No	No	Various Screen Types (Raster vs Vector), Character/Symbol Methods (Dot vs Stroke) and Possibly Color, Peripheral Device Compatibility for Playback
307 - Attitude and Motivation	Operator Performance (Doubtful They Would Use MSRF)	Sonar, AN/SQQ-23 AN/BQQ-5	Early Stages Fidelity not of Prime Importance	None	None	MSRF	None	2	Yes, if Ship's Own Sonar not Available and MSRF has Good Sonar Fidelity	No	
308 - Performance Enhancement	Performance Aids Evaluation Job Designation		Not Very Important	None	Stop for Instructions, Switching of Task	MSRF	None	3-4	Yes	No	
310 - Acquisition and Initial Service											
311 - Design of Manned Systems	Applications of Human Factors to Manned Systems	Sonar, Automatic Bridge, Manning Reduction	Moderate	Video Tape	Only if Problem	MSRF	Audio Recorder, Standard Lab Instruments	8-10	No	Rarely	
312 - Applied Psychobiology	Biological, Psychophysiological Correlates of Performance	Sonar	Not Important	Psychophysiological Data Recording	Unknown	MSRF	Nova EEG Recording System	1 Initially, 2-3 Later Years	Yes, Subject Accessibility	Unlikely	Floppy Disk Compatability with Nova System
02 - Director of Operations	No Use of MSRF										
234 - Computer Support	Computerized Counseling, Adaptive Testing, Efficiency of Data Base, Entry/Query, Graphic Job Mix Analyzer		Not Essential Want 3-D Graphics	Light Pen	None	Shared Own/ MSRF	Access to Outside Computer Systems	1	Desirable, Not Imperative	No	

A consensual impression of the requirements for the MSRF was gathered by considering each column separately.

A primary concern was the Navy systems that each code would be interested in studying. It is clear from the responses under the column labeled "Navy systems" that research interests primarily revolve around three particular types of systems: sonar systems, tactical data systems, and propulsion systems. The requirements for display fidelity range from unimportant to very important. It appears that most research codes would be satisfied with some reasonable but not perfect fidelity of displays for the simulated systems.

Each code was asked to indicate if there were any special equipment requirements that they would like to see available in the MSRF. Audio and video recording equipment was the item mentioned most often. Two codes expressed interest in having the ability to record psychophysiological data. One code indicated a requirement for large-scale random access files, which is more an inherent characteristic of a computer system rather than special equipment. The same is also true for the specification of the availability of light pens for the displays. Most codes had no special equipment requirements.

It is clear from the columns labeled "Experimenter Intervention Requirements" that no code requires continual interaction of the experimenter with the on-going simulation. In most cases the experimenter would intervene only to restart the scenario, give further instructions, or terminate the experiment. Most codes indicated that they would want the MSRF staff programmers to do the applications programming for the research. In two cases the codes desired to do their own programming, and in one case programming was viewed as being a joint effort between the MSRF and the code's staff personnel.

Only three codes indicated any plans to use their own equipment within the MSRF for their experiments. In one case, a NOVA computer which is used for electroencephalogram (EEG)

recording would be brought into the MSRF. In another case, an audio recorder or standard lab instrument would be used. Access to outside computer systems is desired by the Computer Support Department, Code 204. Strictly speaking, this requirement does not fit into the category of equipment brought into the MSRF but would be taken into consideration in the design of the computer system.

A preliminary design guideline for the MSRF was that it be capable of supporting research using up to eight subjects simultaneously. It is clear from the response to this question that most departments would be doing research involving less than eight people, and there were only two indications that more than eight subjects would be involved. The majority of the respondents to the survey indicated that they would desire the MSRF to be mobile if possible. The primary reason given was subject accessibility. Several individuals reported that it was difficult to obtain fleet personnel for subjects because of their work commitments aboard ship. By having the facility mobile and therefore locatable at dock-side, it would be easier to obtain subjects for the planned experiments.

The results of the survey indicated that there would be very little use of classified material in either electronic or hard-copy form used in the MSRF. Since there were at least three indications that classified material might be used, the implications for physical and electronic security would have to be investigated in the design of the facility.

The last question allowed the respondents to provide any additional information or comments on the design of the MSRF. The three codes which provided comments indicated the desire for some means of transferring data from the MSRF computer system to another or vice versa. In addition, one code desired to have various types of CRT display systems available for their research.

After the survey had been completed, a meeting was held at NPRDC to discuss the results. It was evident from the results of the survey that most of the research requirements of the various NPRDC departments could be accommodated in the MSRF. In some instances, it was apparent that it would be extremely difficult to accommodate their desires. For example, the request for various types of display systems has significant cost implications which would place it out of the funding scope for the MSRF. Although there was a general consensus that mobility of the MSRF would be desirable, it was agreed that the MSRF would be a permanent facility at NPRDC. Preliminary evaluation of the requirements to make the MSRF mobile indicated that the overall cost of the facility would be doubled or tripled if it had to be fully mobile. Since the primary reason for desiring mobility was for subject accessibility and not for some clear research need *per se*, it was decided that the subject accessibility problem could be solved by some other means than moving the entire facility.

Knowing which types of Navy systems were likely to be the subject of research initially in the MSRF had important implications for the design. Since the primary research interest was in sonar and NTDS type systems, it was agreed that the initial equipment specifications for the work station consoles would be based on the requirements for simulating such systems. The work station consoles for these two systems are sufficiently elaborate that the equipment requirements for other systems would be a subset of the equipment required for sonar and NTDS work station consoles. Display fidelity was considered of sufficient importance that a great deal of effort would be spent on specifying a display system which would meet almost all research requirements with the exception of those requirements for exact fidelity of a sonar display.

Several codes were interested in having video recording systems available in the MSRF. Since several of these systems

are available at NPRDC, it was decided that no additional systems would be purchased for the MSRF, but that one or more of these systems would be made available for use within the facility. It was decided that all of the other research requirements expressed in the results of the survey could be easily accommodated in the MSRF design and impose no particular problems.

In general, the results of the survey confirmed the original expectations about the purpose and use of the MSRF. The survey also tended to confirm that the MSRF would be very useful to most of the research-oriented departments within NPRDC and would be capable of supporting their research needs.

Design Philosophy

GENERAL GUIDELINES

Beside providing the required support for the research that is anticipated to be conducted in the MSRF, four general guidelines were imposed for the design of the facility. These guidelines are applicable primarily to the MSRF computer system and the work station consoles.

In keeping with the intent of the MSRF to support a wide variety of research on man-machine interaction in Navy systems, the MSRF's research support capability must be flexible. That is, it must be relatively easy to change the application software and the appearance and operation of the work station consoles to meet different research goals. Flexibility implies that the computer system software and work station consoles should have a high degree of modularity. Thus, if additional software routines or equipments need be added or deleted for particular research purposes, it should be easy to do.

An important consideration for flexible design is the amount of computer processing power required to support the

anticipated research. If a computer system with a single central processing unit (CPU) was designated for inclusion in the MSRF, it would have to have sufficient power to support the most complex experiment that is anticipated to be conducted in this facility. If the processing power required subsequently exceeded the capability of the designated computer system, it would be necessary to upgrade the computer system, i.e., replace the then current system with a more complex and expensive one. Clearly this would be an undesirable state of affairs, particularly since the vast majority of research would probably not require the same power as the most complex research project. A better approach than depending on a single CPU system for supporting the MSRF research would be to use a distributed processing system. That is, rather than a single large computer, the use of several small computers, linked together, would allow the processing load to be distributed among the various CPUs. Also, additional processing power could be added in a modular fashion. Therefore, a distributed processing approach to the MSRF computer system appears to be very desirable in keeping with the general guideline of flexibility.

Closely related to the concept of flexibility is the provision for growth of the MSRF. That is, as experience is gained by the conduct of research in the facility, it will probably be desirable to do research with increasingly larger numbers of subjects. The original intent is to provide support for research on up to eight individuals working at up to three work station consoles. In future years it may be necessary to add additional consoles. Therefore, the computer system including both the hardware and software must be open-ended to allow for growth.

It is anticipated that the MSRF will be developed over a period of approximately 3 years. The first year's acquisition must include sufficient equipment and software development to

allow research to begin. Additional equipment and software would be added during the following 2 years of development. Having an initial research capability early on will be valuable not only in terms of using the available capabilities of the MSRF but also because it is highly likely that lessons will be learned about what types of additional equipment and software development would be desirable during the future years of development. If the MSRF could not be used until its full maximum capabilities have been developed, it is possible that certain functions or equipment would be found to be inappropriate or inadequate. A good deal of money could be spent for multiple copies of high-cost items such as the display hardware which may later prove to be inadequate in some way. Planning for initial research capability will ensure that the functional features of the MSRF equipment and software are satisfactory for their intended purpose before additional procurements are made.

The last general guideline is that the MSRF should have as low a cost as possible consistent with the primary objective of supporting the intended research and that the cost should be distributed as evenly as possible over a period of 3 years. Although minimum cost is an objective of every Government procurement, it has particular relevance to the MSRF. Most facilities or equipments procured by the Government are based on minimum cost consistent with meeting explicit performance objectives. In the case of the MSRF, the performance requirements are not entirely specific, and therefore a certain amount of judgment is required in deciding what capabilities are required and how they will be achieved. For example, the display system to be used in the MSRF will be the single most expensive item next to the computer system. Since the anticipated research projects require some fundamentally different types of displays, it would be relatively easy to find a variety of display systems, each of which was applicable to a

particular system simulation. However, by developing a single display system which meets almost all requirements for all types of anticipated research, a considerable savings might be realized. To meet the objective of providing the required capabilities for the MSRF at the lowest possible cost, every attempt has been made to specify equipment that has the greatest versatility for supporting a variety of research requirements. Also in order to spread the costs of development of the MSRF over 3 years, only the minimum necessary equipment and software development needed to provide an initial research capability will be specified for the first year's acquisition.

SPECIFIC APPROACH

In order to meet the specified research requirements and the general guidelines for development of the MSRF, the conceptual approach to design was based on four principles: software intensiveness, modularity, maximum use of off-the-shelf items, and versatility of function for both the software and equipment.

Software intensiveness means that all aspects of the simulated system and experimental requirements such as data gathering are under control of the computer system and all functions of the display and the input devices are routed through the computer system. Therefore, under this approach, use of devices with "built-in" functions is minimized. Also, most information to or from the work station consoles can be routed throughout the entire system under software control. Rewiring should never be necessary to reroute information or change the function of any device. In a few cases, patch panels will be used for rerouting information.

Modularity of software components and physical devices is one component of obtaining overall flexibility in the MSRF. The software will be modular in the sense that each component should perform a definite function with as much generality

of application as possible. Also, by specifying a convention for communications between modules, systems and applications programs can be synthesized in a building-block fashion with only sufficient additional software required for specific details of the application. During the development process of applications programs, attention must be given to writing them in as general a form as possible so that they have potential use in future applications.

Equipment modularity refers principally to the components of the work station console. As will be seen, the framework of the consoles themselves can be reconfigured to any desired shape or size. The devices which will be incorporated in the console will be packaged in mounting frames which will allow them to be placed in any desired location within the work station console. Also, the interconnection of the devices to the computer system will take place through a common interface panel. The devices connected to this panel will have common electrical characteristics to the greatest extent possible so that a given device can be replaced by a different device of similar function without the necessity of cutting or soldering wires. For example, it may be desirable for one type of research to use a joystick for cursor control. In another system, the more typical cursor control device may be a trackball. It should be a simple matter to substitute one device for the other with very little effort.

Use of off-the-shelf equipment, i.e., devices readily available through commercial sources, will have great benefits in terms of the simplicity and time required to set up a particular experiment. Also, maintenance problems will be minimized since, as a last resort, the device can be returned to the manufacturer for repair. Use of specially built devices has therefore been minimized.

Since all systems that will be simulated are Navy systems, the equipments actually used in the real systems usually meet

military specifications. For research purposes, it is not necessary to incur the additional cost of procuring equipment that meets military specifications since the equipment will not be subject to the environmental stresses that militarized equipment is expected to withstand. In many cases, commercial grade versions of militarized equipment are available. In cases where they are not, usually a commercial device of functional equivalence can be found. Therefore, equipment that meets military specifications has been excluded as a requirement for the MSRF.

Use of multipurpose equipment whenever possible was part of the design approach. For many research purposes, it is necessary only to provide functional equivalents rather than exact duplication for the display and control devices. In some cases a device has been specified that is more expensive than any of several devices which each have a specific application. For example, several system consoles in actual Navy systems use multifunction button arrays. Each button has several legends which can be projected depending on the state of the system. Generally, each button is limited to, at most, 12 or 24 legend options. The size of the button matrix varies depending on the system in which it is incorporated. Rather than specifying several types of multifunction buttons, plasma touch panels have been substituted. These panels have a higher cost than any particular variable legend button matrix. A plasma touch panel, however, is a much more versatile device. Any desired legend can be displayed on the panels, either with alphanumerics or symbolically, and "virtual" buttons can be positioned (drawn) anywhere on the panel. This device will be described in greater detail later. The important point is that the plasma touch panel has a much greater potential applicability for a variety of simulated consoles than any given fixed-button matrix. Also, the size and shape of the buttons and the legends displayed are completely under software control and do not require change of film chips to change legends.

The plasma touch panel is therefore compatible with the concept of software intensiveness since the device can be reformatted under software control.

These four key elements in the design approach ensure that the general guidelines requiring flexibility, potential for growth, an initial research capability early in the development process, and minimum cost are met. Incorporation of these principles in the design will be seen as each specific element of the MSRF is discussed in detail.

SUBJECT WORK STATION CONSOLES

This section specifies the functional requirements and other design features of the MSRF subject work station consoles, and describes the considerations from which these requirements and features were determined.

Background

The display console is essentially a man-system interface. It provides system information to an operator and accepts control inputs from him. Older types of consoles are characterized by a relatively large number of discrete information sources, and a large variety and number of control inputs. The reason for the seeming complexity of the older display consoles is that usually a great deal of information could not be presented in an integrated fashion on a single display device such as a CRT. Also, control inputs affected various discrete system components and therefore the type and number of controls required could vary greatly depending on the system. Different consoles for different systems were generally built by different contractors. Consequently, although many of the functional requirements of the consoles were the same from system to system, the appearance of the consoles varied greatly. Over the years, the trend in display console design has been to integrate more and more display and control functions and thereby reduce the number of discrete devices required on a console.

The most modern display consoles have a limited number of display devices and a relatively small number of controls usually of the button type, whose function varies depending on the mode of operation of the console. An example is the UYQ-21 console, currently being produced by Hughes Aircraft Company, which is the most modern display console that will

be incorporated in Navy surface ship systems. This particular system will be described in greater detail later.

The major difference between the older display consoles and the modern display consoles is that the new consoles are primarily man-computer system interfaces. The operator, in effect, is communicating with a single device, a computer, rather than a multiplicity of devices. Advances in display processing technology and the trend to preprocess and integrate information by computer have allowed the integration of information from a variety of sources into one or two displays. Control inputs go directly to the computer and either affect the computer operation directly or are translated by the computer into the appropriate signals for other devices, such as sensors, connected to it. Only simple means of communication to the computer are required, such as buttons, since the physical input signal can be translated to a variety of output signals. Also, fewer buttons are required, since the computer, at the operator's request, can alter the functions of the buttons.

Older console display systems had a variety of display devices. Usually a central CRT was used to provide sensor information, such as sonar or radar, and numerical information was provided on auxillary displays. Developments in information display have allowed the combining and integration of information into a single display. Most information displayed on modern consoles has been preprocessed, transformed, and integrated with other information in a unitary computer-generated display which is presented to the operator. The amount of information that can be displayed on a CRT is directly related to the sophistication of the data- and image-processing capabilities of the computer and display system. As would be expected, the greater the sophistication of the display system, the higher the cost of the system. This fundamental trade-off is the primary problem for the design of the MSRF console. The console must be capable of reproducing the

informational content of modern sophisticated consoles at a cost less than actual system consoles. Considerations of the MSRF display system will be discussed in more detail presently.

Major Systems for Simulation in MSRF

The previously discussed survey indicated that most research interest would center around Naval tactical data systems (e.g., NTDS) and sonar systems. Some interest was also expressed in bridge control, electronic countermeasures, and propulsion systems. The results of the survey were reviewed in a meeting with NPRDC personnel and it was decided that the MSRF consoles should be primarily designed to simulate NTDS and sonar systems. The functional requirements for these two types of systems will be discussed in some detail. The remaining systems identified for possible research in the MSRF will be discussed only in general terms.

NTDS

Table 2 lists the fundamental functional requirements for an NTDS console. The console currently used for NTDS systems on surface ships is the UYA-4. A schematic representation of the UYA-4 console is shown in Figure 1. Table 3 identifies the major display and control features of the console. The letters on the figure correspond to the table.

Figure 2 shows a schematic representation of the UYQ-21 console in the NTDS configuration. This console is scheduled to replace the UYA-4 in the near future. Table 4 identifies the major features of the UYQ-21 console. This console performs all the functions listed in Table 1. Note, however, that the display and control features of the UYQ-21 shown in Table 4 are about half as many as for the UYA-4 shown in Table 3. The number of discrete elements for information display and control inputs have been greatly reduced in the UYQ-21. This has been possible largely due to developments in the area of computer control and display processing.

TABLE 2
NTDS CONSOLE FUNCTIONS

A. INFORMATION DISPLAY
   1. Raw Radar
   2. Superimposed Symbology
   3. Amplifying Data Readouts
   4. System Alerts
   5. Time To Go Indicator

B. CONTROL FUNCTIONS
   1. Fixed Functions
   2. Mode Selector
   3. Radar Selector
   4. Range Scales
   5. IFF/SIF and SIF Gate Functions
   6. Category Select
   7. General and Special Purpose Function Codes
   8. Cursor Control
   9. CRT Tuning

C. COMMUNICATION
   1. Interior Communication Channels
   2. Exterior Communication Channels

SONAR SYSTEMS

Table 5 lists the fundamental functional requirements for a generalized sonar console. Not all sonar systems currently in use employ all these functions. Some employ a limited number of these functions, whereas the more modern systems can perform all the functions listed. Most sonar systems pertinent to the MSRF console design are discussed below.

Submarine Sonars

BQR-2, BQR-7, BQS-6, BQS-13. These older submarine sonars are characterized by one- or two-console work stations employing mechanical numeric displays, electrostatic paper graphic

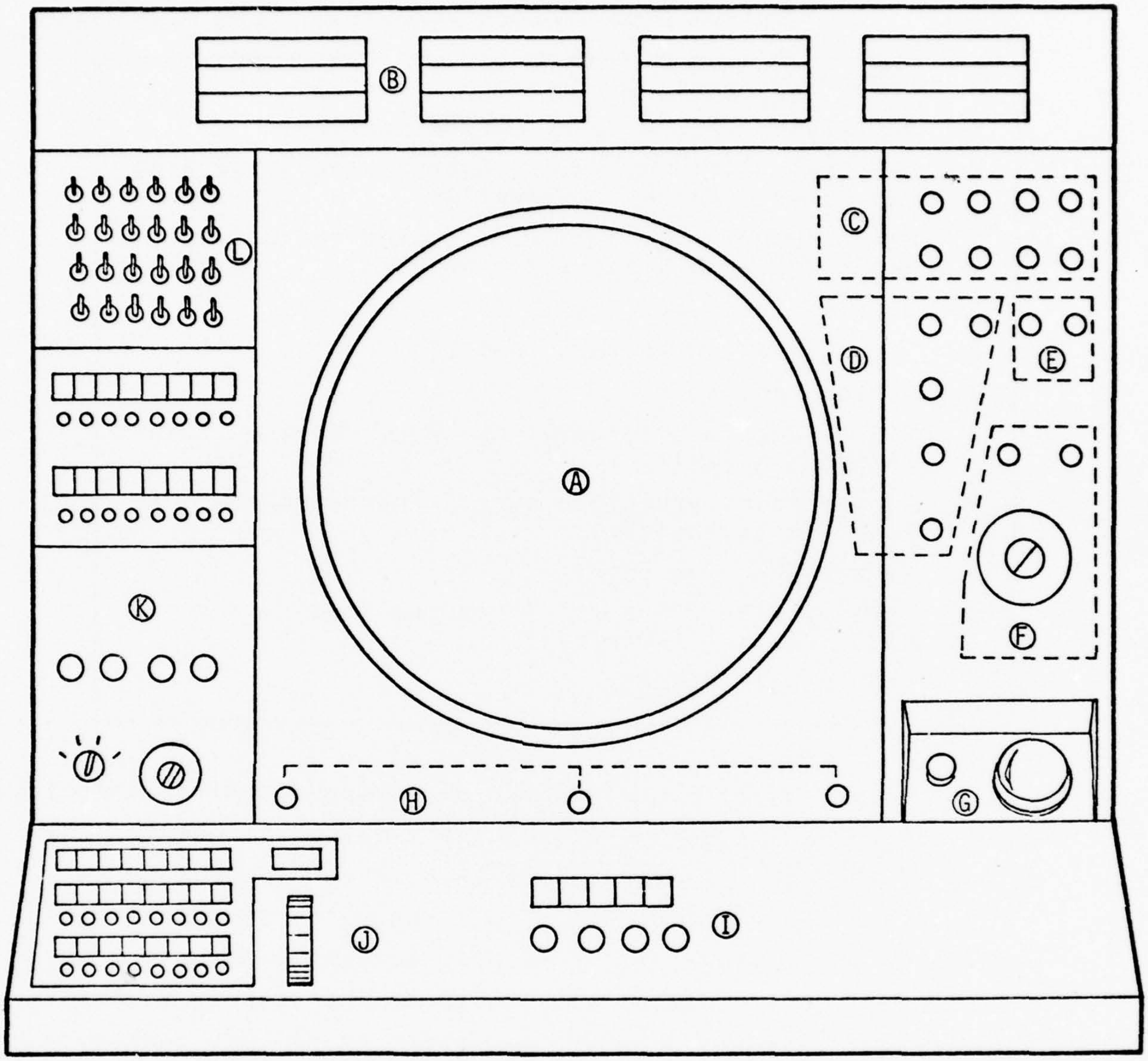


Figure 1. Schematic representation of UYA-4 NTDS console. Circled letters correspond to entries in Table 3.

displays, fixed-function push buttons, and, in the case of the BQS-6 and -13, a CRT display for PPI sonar data only (no superimposed symbology or alphanumerics).

BQQ-5. The BQQ-5 is a new sonar which will be backfitted to existing attack-class submarines (SSNs) to replace the sonars mentioned above, as well as being employed in new-construction

TABLE 3
UYA-4 NTDS CONSOLE FEATURES

A.	Radar and Symbology Display
B.	Amplifying Data Display (Alphanumerics)
C.	CRT Tuning (Video, Sweep, Symbols and Range Ring Brightness, Focus, Astigmatism)
D.	Radar Function Control (CRT Center and Off-Set, Radar Type and Range Scale Select)
E.	SIF/IFF Challenge Functions
F.	Vehicle Motion Vectors Control
G.	Cursor Control
H.	Fixed Functions (Drop Track, True Bearing, Enter Mode and Radar)
I.	Number Entry (Function Codes, Track Number, SIF Code, Target Height)
J.	Variable Action Entry
K.	Communications Control (Intercom Station and Radio Channel Selectors)
L.	CRT Symbol Editing

SSNs. It is a multi-console sonar employing identical consoles consisting of two similar rectangular CRT screens mounted one on top of the other, variable-function buttons, and a force-stick cursor control. Figure 3 is a schematic representation of the BQQ-5 console (the top display screen is not shown in the figure). The apparent simplicity of the console is notable. However, this console permits the selection of very many more operating mode parameters and display formats than any of the older sonars which it will replace. Each variable-function button in the control matrix has up to 24 rear-projected legends (and corresponding functions) which appear depending upon the state of the system at any given time. Sensor data may be displayed in a variety of formats, together with cursor indicators, superimposed tactical symbology, and alphanumeric target and system information, format legends, etc.

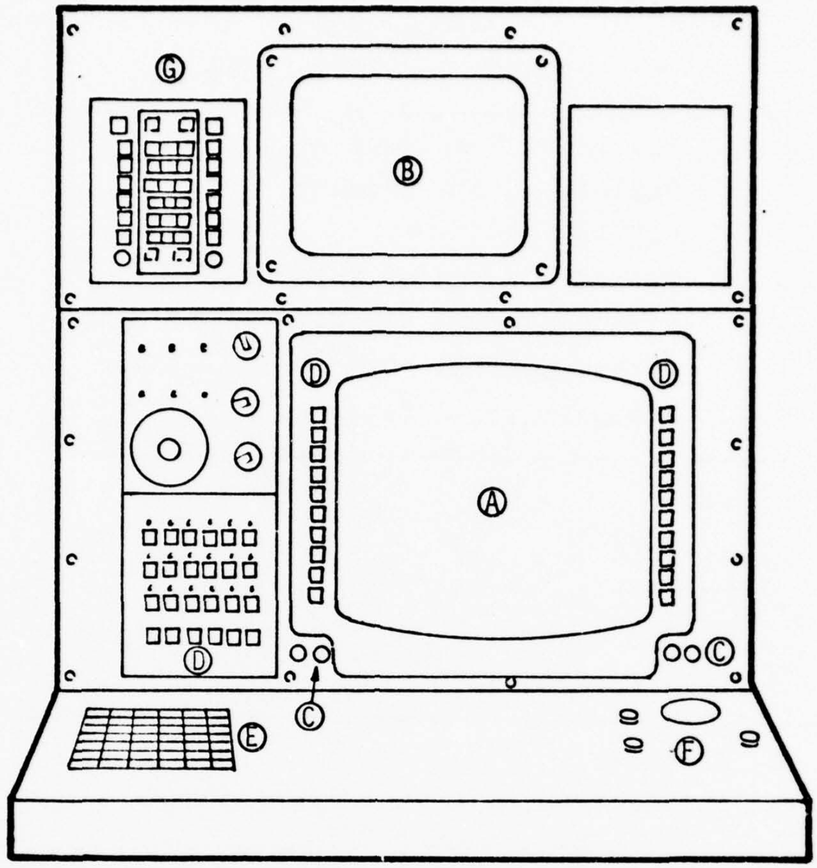


Figure 2. UYQ-21 console in NTDS configuration.

BQQ-6. The BQQ-6 is functionally similar to the BQQ-5. It will be employed in the new TRIDENT program Fleet Ballistic Missile submarines.

BQR-21. The BQR-21 is a new sonar which will be backfitted to existing Fleet Ballistic Missile submarines to replace the older BQR-2 and BQR-7. It is a single-console sonar with three multipurpose CRT displays and digitally controlled multifunction push buttons.

Surface Ship Sonars

SQS-23, SQS-35, SQS-56. These are relatively simple surface ship sonars which employ single consoles with single fixed-format CRT PPI displays of sensor data (and in electrostatic paper graphic display in the case of the SQS-35), and fixed-function controls. The SQS-23 is a relatively old

TABLE 4
UYQ-21 NTDS CONSOLE FEATURES

A.	Radar and Symbology Display
B.	Amplifying and Readout
C.	CRT Tuning
D.	Variable Function Buttons
E.	Fixed-Function and Numeric Buttons
F.	Cursor Control
G.	Communications Selector

TABLE 5
SONAR SYSTEM REQUIREMENTS

I. DISPLAYS
   A. Active
      1. PPI
      2. A-Scan
      3. B-Scan
   B. Passive
      1. Time-Bearing Recorder
      2. Narrowband
      3. Broadband
      4. Multiple Simultaneous Displays of Various Integration Times
   C. Symbology
II. CONTROLS
   A. Mode
   B. Pulse Length
   C. Range Scales
   D. Cursors
   E. CRT Tuning

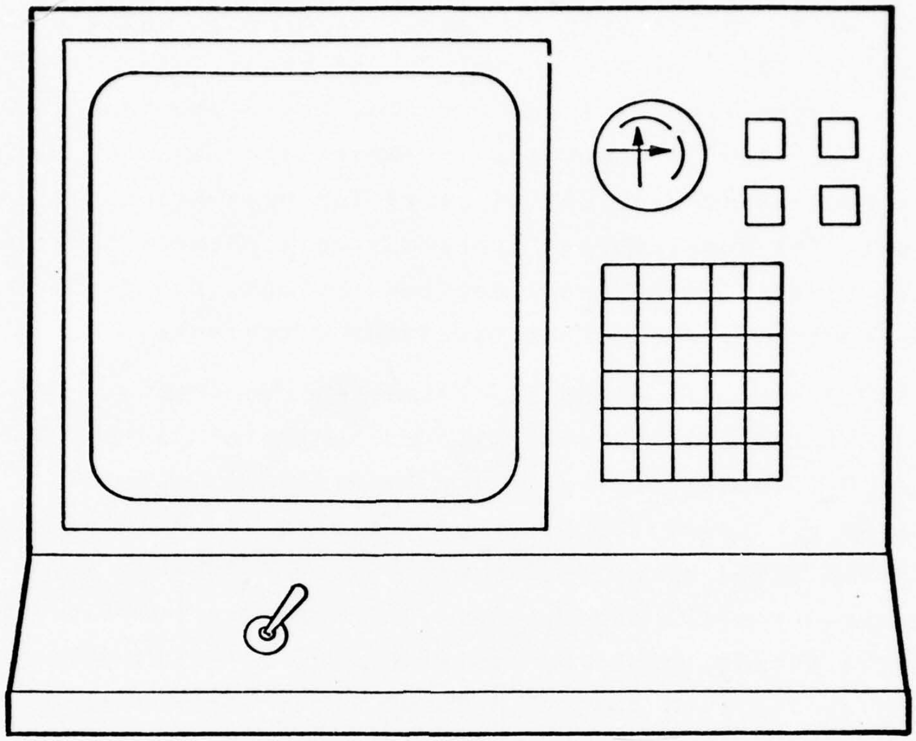


Figure 3. Schematic representation of BQQ-5 sonar console (second display on top of console not shown).

hull-mounted sonar. The SQS-35 is a newer variable-depth sonar which can be expected to be in service for some time. The SQS-56 is a new hull-mounted sonar which will be employed in the new-construction FFG-7 class.

SQQ-23. The SQQ-23 is currently used but is not common. It is the first sonar to use digital signal processing to allow variable display format. The system itself consists of two consoles with one main display screen each. The two consoles are separated by a control console which contains six function switches for changing the display format. The change of display format is under the control of the sonar supervisor. A paper recorder is located on the top of each display console.

SQS-26, SQS-53. The SQS-26 is a long-range, hull-mounted surface ship sonar used throughout the FF-1052 class and in a few other ships. Thus, it is a very common surface ship sonar.

The SQS-53 is essentially an SQS-26 with modifications to make it suitable for use in the DD-963 class. The SQS-26 and -53 systems employ three operator consoles, two of which are very similar in appearance. These main consoles each employ two raster-scan CRT displays for presentation of sonar data only (no superimposed symbology or alphanumerics) in a fixed format, with fixed-function buttons, force-stick cursor controls, and mechanical numeric readouts.

SSSMP, SQR-19, LAMPS III Shipboard Processing. None of these advanced surface ship systems is in the Fleet yet. The Surface Ship Sonar Modernization Program will modernize the SQS-26 by replacing the signal processing and display units. The SQR-19 towed array system will provide surface ships with an improved passive sonar capability, and the LAMPS III program will permit shipboard processing of acoustic data from helicopter-deployed sensors. These three sonar systems will have in common the use of the UYQ-21 acoustic display console. The UYQ-21 display console has been designed to provide a standardized approach to the display requirements of the surface Navy. The console was designed to meet needs of several shipboard applications including NTDS, sonar, fire control, and electronic warfare. The UYQ-21 console in the sonar configuration is shown schematically in Figure 4. Like the NTDS configuration, the sonar configuration is functionally very simplified compared to older systems, although it has much more versatile capabilities. The basic elements of the UYQ-21 are CRT displays, a trackball cursor control, a 6x6 fixed-function button matrix, CRT controls, and two columns of variable function buttons mounted on each side of the CRT. The labels for these variable function buttons appear on the CRT screen and are thus under software control. The UYQ-21 will be the most common and versatile display console for future use in Navy surface ship systems. Because of this, we will examine the characteristics of the UYQ-21 display system in greater detail in a later section.

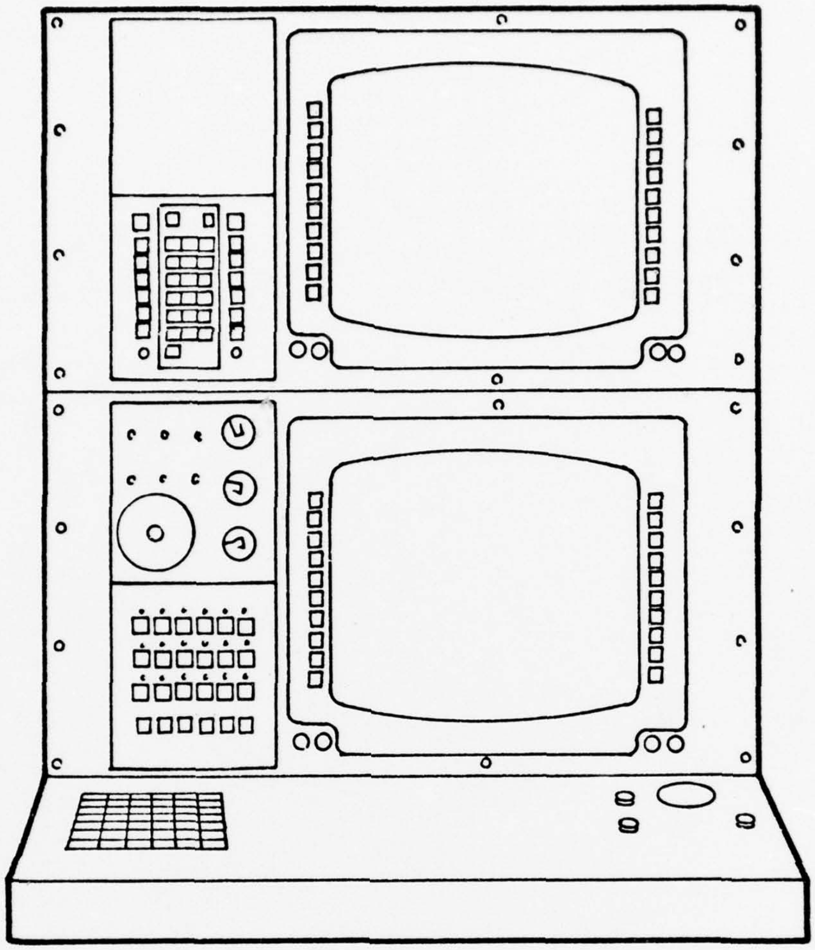


Figure 4. Schematic representation of UYQ-21 sonar console.

INTEGRATED BRIDGE SYSTEM

A schematic representation of the integrated bridge system is shown in Figure 5. Table 6 details the functional characteristics of the labeled devices on the integrated bridge console. It has a general similarity to NTDS and sonar consoles in that it has a central CRT display, a small alphanumeric CRT for data related to radar contacts or navigation, a variety of single-function push buttons, and a cursor control. A small additional LED display gives numeric information for course, speed, range, and bearing. Like most modern Navy display

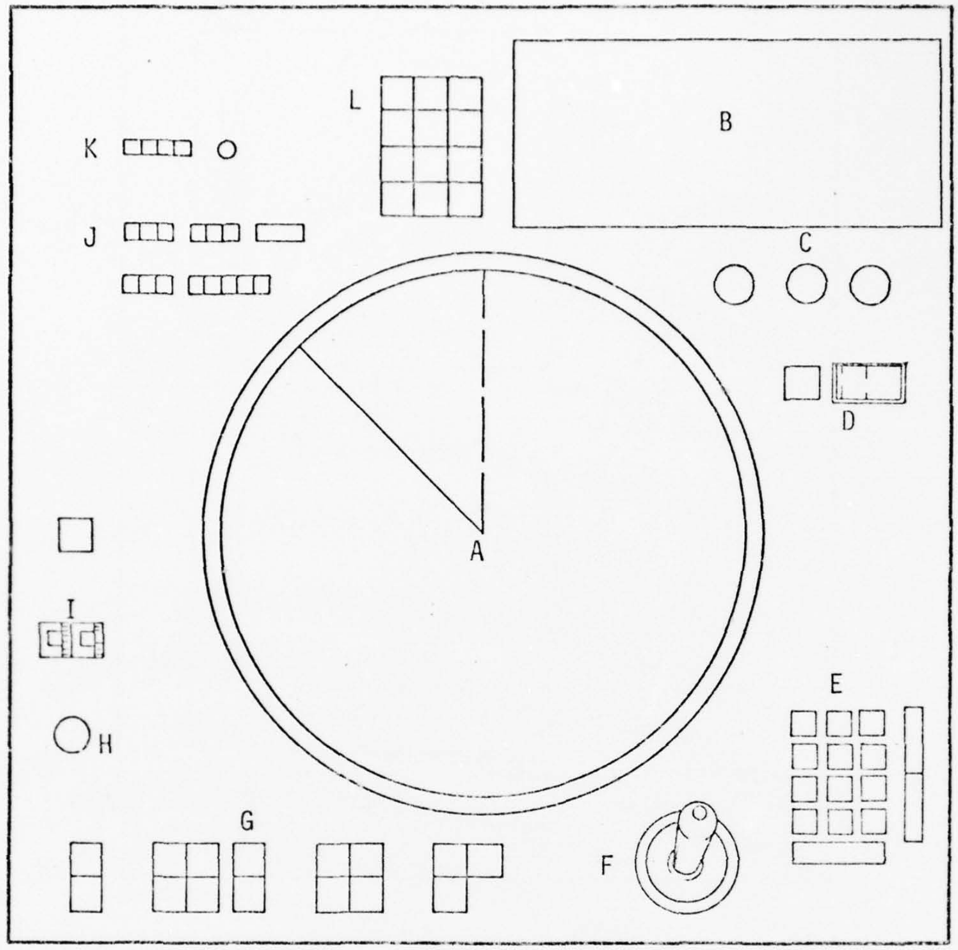


Figure 5. Schematic representation of main console of the integrated bridge control system. Letters correspond to entries in Table 6.

consoles much of the information presented on the primary and secondary displays is computer processed and control inputs are communication links to the computer rather than to discrete electrical devices.

PROPULSION SYSTEMS

Propulsion system consoles are very different in function and appearance from the display consoles just discussed. They consist primarily of two types. The older type of propulsion console consists of needle-type indicators, discrete push buttons and rotary knobs, and large hydraulic control valves.

TABLE 6
INTEGRATED BRIDGE CONTROL SYSTEM FEATURES

A.	Primary CRT Display - Radar and Symbology
B.	Amplifying Data CRT - Contact or Navigation
C.	CRT Tuning
D.	Display Information Source Control - Computed or Actual
E.	Numeric Keypad Data Entry
F.	Symbol Cursor Control
G.	Navigation and Maneuvering Controls
H.	Bearing Cursor Control
I.	Trial Speed Entry
J.	Navigational Data Display
K.	Time Indicator and Set Control
L.	Edit and Data Request Entry for Amplifying Data

The second and more modern type has the appearance of a schematic diagram of the propulsion system. Information about various system functions is displayed on an alphanumeric CRT, and various points on the schematic map have indicator and warning lights. Discrete panel meters and dedicated function switches are also employed.

Despite the apparent uniqueness of these consoles, the hardware/software requirements to *simulate* a modern propulsion system console are largely a subset of those required to simulate NTDS and sonar consoles.

IMPLICATIONS

It is clear without much analysis that the MSRF subject work station consoles must provide for graphic display systems, operator controls, a physical framework, and means for electrically interfacing and controlling the console components, both within the console and with an external computer. An in-depth analysis of the major systems for simulation in MSRF, including all those discussed in the preceding overview,

indicated that the graphic display system would be the most costly element of the MSRF console, and the one most difficult to specify to achieve an optimum trade-off between capability and cost. There presently exists a remarkably diverse display technology which had to be reviewed to make an appropriate selection. Because of the central significance of the display system to MSRF, both functionally and economically, we feel it would be well to summarize current display technology before examining the rationale that leads to our specific recommendation.

Display Technology

IMAGE-PRODUCING TRANSDUCERS

The easiest modules to describe are the physical devices capable of producing a visible image. All of the devices of interest to us generate images as luminosity variations on a screen of some sort; the principal differences between the various technologies represented are between their intrinsic image characteristics and between the nature of input each requires.

Intrinsic Memory Devices

There are several display elements in this category. Their common characteristic is that each may logically be regarded as a device capable of storing an image as well as displaying it. The major distinction between them is the extent to which the stored image may be altered.

Matrices of discrete luminous elements. Devices in this category produce a picture using a pattern of dots, squares, or rectangles. The display surface is normally organized as a rectangular matrix of these elements with fixed sizes, spacings, and positions. Each element (often called a pixel, from "picture cell") in such displays must usually be in one of two luminance states, corresponding to light or dark. Therefore,

images possessing more than two shades of gray must be represented by halftone techniques.

Display quality for these devices is almost wholly determined by the density of their matrices in units of dots per degree of visual angle. This density directly controls the resolution and reproduction of tones in halftone images. It defines the minimum sizes of objects whose shapes may be drawn unambiguously. Perhaps more importantly, whenever a straight line is drawn at an arbitrary angle across such a quantized space, the line will be visibly non-straight except at extremely high densities due to inexact mapping onto the matrix. At normal viewing distances, this effect is quite noticeable even at densities as high as 50 pixels per cm. Typical devices have densities on the order of 25 pixels per cm. For a given display area, price is usually a function of the square of density.

To enhance the practicality of these devices, each pixel usually behaves as an autonomous memory element which may be set light or dark without disturbing the remainder of the display. How this is implemented depends on the device. The most primitive type of input required is usually in the form of a stream of encoded digital commands, each of which provides the address of a pixel and an indication of the state to which it is to be set. Some devices support more complex commands for such purposes as erasure of large areas or alteration of several pixels with a single command.

LED matrices: Some manufacturers produce closely spaced arrays of light-emitting diodes mounted in opaque panels with integral memory devices. Initially, the available products only contained partial matrices suitable for displaying alphanumeric characters in fixed positions with no intervening information. However, the usefulness of such devices in providing hand-held graphics displays is motivating the production of fully filled matrices. The principal area of application for these relatively inexpensive devices would seem to be the implementation of small-size

message display areas, variable function touch panel labeling, or perhaps small-size virtual metering devices. Their use in large displays intended to be studied *intently* by operators for extended periods may be limited by "eye fatigue" resulting from the near-coherence of the light output.

Plasma panels: These devices are somewhat more expensive than LED arrays, but seem to be capable of serving satisfactorily in more application areas. They consist of transparent panels, available in a variety of sizes, within which exist matrices of regions wherein gas discharges may be activated or deactivated individually. Each region, or pixel, acts intrinsically as a one-bit memory device. The panels emanate light of a color similar to that of neon light bulbs, and their widespread use in terminals suggests that operators can tolerate extended periods of their use.

The transparency of the plasma panels represents a unique capability among transducers: they are the only luminous display screens we know of which may have another image projected through them (there exist CRTs containing projection ports, but these CRTs are extremely special-order devices and are not incorporated in any general-purpose systems of which we are aware).

Direct-view storage CRTs. These devices have served many purposes over the years. Before their appearance in computer displays, their most familiar use was in special oscilloscopes where they captured records of fleeting, nonrepetitive phenomena otherwise invisible. When compared with other types of transducers, they tend to come out as either the very best or very worst, depending on which characteristic is being considered.

In general, storage CRTs maintain an image memory in the form of a pattern of charges on an internal grid. This pattern modulates the intensity of a wash of electrons which then strikes the phosphor on the screen, producing luminosity. A deflection system, normally operating from analog parameters X and Y, controlling beam position in rectangular coordinates,

and Z, controlling beam intensity, is used to place charges on the storage grid. A picture is built up by incrementally illuminating areas on the grid. While the physical properties of this grid do, in reality, imply some quantization of beam position, one may for practical purposes regard placement of points of light on the screen to be continuously variable. A significant point often overlooked by persons defining the specifications for display systems is that this attribute of continuous point placement is of extreme importance in the quality of a display's appearance. For example, some storage CRTs have no better "resolution" than do plasma panels--yet the difference in display quality can be staggering when one looks at such objects as circles drawn on each.

While it is possible to achieve limited gray scale on some storage CRTs, it is much simpler to adjust them for bistable operation and achieve shading, if necessary, through halftones. Some storage tubes bring with them other limitations; there may be limits on the percentage of illuminated surface within an area of a given size and the image usually decays over time. In some cases, there exists a time limit for retention of an image beyond which CRT damage may result.

The main defect of storage CRTs in many application areas is that the only practical dynamic alteration of their displays is addition of new illuminated areas. Erasure is usually total, takes up to 1/2 second, and normally produces a highly obtrusive flash on the display.

As a class, the intrinsic memory devices offer the *least expensive* path between a digital data base and a graphic display. They may be configured with very little supportive hardware. Since the image, once composed, is stored entirely within the devices, extremely complex displays may be produced by *extremely small and simple systems*. The images produced are *stable and flicker-free, regardless of complexity, and the attainable complexity is extremely high*. For example, when

the information content of a display is measured in bits, it may be said without reservation that a 512-by-512 element plasma panel can display 262,144 bits worth (2^{262144} unique images possible), at a cost on the order of 2¢ per bit. The cost of a storage CRT display of comparable size and resolution is about the same, but the price per unit information is somewhat less since continuous variability of spot position gives a CRT display an intrinsic advantage in information content. LED matrices are considerably less expensive in the small sizes currently available, but their price per bit is actually higher than that of plasma panels.

The main weakness of intrinsic memory devices is a natural consequence of the very fact that they provide their own image memories. This weakness becomes immediately apparent when one tries to use them to implement dynamic displays. As stated earlier, storage CRTs may only be altered by *addition of bright points* to an image being displayed, which may at least be done smoothly or by *total erasure and reconstruction of the altered image* which, with its bright flash and long duration, totally disrupts any continuity one may wish to convey in presentation of the dynamic display. In fairness it must be conceded that some storage CRTs are capable of superimposing dynamic, refreshed images on the stored images but presently these are necessarily of low brightness and limited usefulness.

The discrete-matrix devices fare somewhat better but can present serious difficulties, depending on the nature of the image's topology and dynamics. The trouble arises whenever two different entities being depicted occupy the same space on the screen. A simple example might be two crossing lines. At the point of crossing, there is usually at least one illuminated pixel which in fact represents a point on *both* lines. Now suppose that we wish to move one of the lines. This is normally accomplished by erasing the line and then redrawing it in its new position. Yet if we erase all the pixels on

that line, we will also erase the pixel(s) it *shares* with the line it intersects, thus leaving a gap in the remaining line at the previous point of intersection. The more the dynamic parts of a display are moved, the more this erasure causes the static parts they contact to vanish.

The result often negates the system-simplifying effects of intrinsic image memories. One might think that, by inspection of the image memory, he could detect these points of intersection and simply skip over them. However, since each pixel's memory contains only one bit of information, such an inspection must rely heavily on inference and there does not exist a general solution which uses only the data contained in the image memory. The solution normally employed with such devices to achieve satisfactory dynamics is to erase the display and redraw it in altered form. This implies two things. First, the complexity of the display and the speed with which it can be written determine the extent to which visible flicker may occur. Second, it becomes necessary to maintain somewhere within the system an up-to-date description of the display in terms of pictorial entities, or *graphic elements*, rather than a pixel-by-pixel representation of the display. How and where this description is maintained determines the extent to which it burdens the display system's resources, but the requirement of recomposing the entire picture every time something moves can safely be characterized as a major increase in expense or loading however it is done.

Directly Viewed CRTs

The venerable cathode-ray tube is the transducer of choice in most existing display systems. The designer has a rich field of choices in most important physical dimensions, such as size, shape, phosphor color and persistence, resolution, spot size, luminance, signal-to-noise ratio, and deflection speed. Pictures may be created in a variety of ways, depending on the deflection system chosen.

As a transducer, the CRT has very few inherent limitations beyond those implicit in its physical characteristics. As we shall see in our discussion of deflection systems, it is feasible to display anything within the capability of intrinsic memory devices on a normal CRT (with the possible exception of certain extremely dense storage-CRT displays and through-projected images on plasma panels).

With this flexibility, however, comes a serious burden on the display system. Images written on a CRT decay rapidly, implying (1) that the display must continually be rewritten or refreshed and (2) that somewhere within the system there must exist some element which either continually generates or retains a memory of the information in the display. The method chosen for performing these two functions is the determinant of capability for a CRT display.

Beyond matching the capabilities of intrinsic memory devices, CRTs are capable of two other important feats. One is the ability to display continuous gray scales within the dynamic range of the particular CRT. The other is the ability to display color images. While these two capabilities are not in widespread use for computer-generated displays, they are critical to many familiar systems such as television, radar PPI, sonar, and other military systems. The expense of such systems has prevented widespread exploration of their possibilities and, in the case of color displays, there even appear to be some relatively untouched human engineering research areas. Nevertheless, it may be expected that as these capabilities become more widely available the using community will discover important means of representing information more effectively by their use.

CRT DEFLECTION SYSTEMS

Deflection systems are modules which interface well-defined analog input signals to the CRT transducers. Cathode-ray tubes produce images by stimulating phosphors with an

electron beam. This operation is controlled by an absolute minimum of three variables: the intensity of the beam and two variables defining position on its two-dimensional display surface. While other organizations are possible, the majority of CRT systems control position by deflection on two orthogonal axes, conventionally called X and Y. This is the simplest deflection system, and is known as X-Y-Z, where Z is the intensity variable. Typically, the driving circuitry for the CRT will include compensation for various nonlinearities and distortions peculiar to the devices so that the X and Y inputs may be treated externally as pure rectangular coordinates for a point on the screen.

Thus, the input to an X-Y-Z CRT deflection system is in the form of three independent, *precisely synchronized* analog signals. X-Y-Z deflection imposes less direct restrictions on the capabilities of a display system than does any other system in widespread use, quite naturally because it embodies a minimum of assumptions. The system can be directed to illuminate any desired point on the display at any desired level at any time. The only limitation imposed if flicker is to be avoided is the combination of bandwidth on each of the three inputs as combined with the persistence of the CRT phosphor.

With this generality come two main penalties. By requiring three channels of information input, an X-Y-Z deflection system needs to receive information at a high rate. This can impose heavy requirements on the module charged with generating those inputs. More important, however, is the electrical problem involved. Each of the inputs has a high bandwidth and all three must be closely in synchronization. These two problems require special attention in transmission and switching of the input signals.

A second major type of deflection system alleviates these problems with some sacrifice in generality. Known as raster-scan

or video deflection systems, these require only one channel of information. The beam is moved across the CRT face by internal circuitry in a *fixed* path which results in sufficiently complete coverage of its surface at some fixed rate. The only input signal is "Z," controlling the intensity of the beam. Any source providing this signal must be aware of the beam path created by the deflection system and must provide an intensity value for each point on the surface at the same time as the deflection system is positioned at that point. This synchronization may be achieved by several methods; most frequently, some single module in the system generates a pulse train which is distributed to the other modules of the system and which is used to define reference points in time. Sometimes the pulse train is combined with the intensity signal to form *composite video*.

While this scheme is conceptually more complex and less direct than random X-Y-Z deflection, it is highly practical and relatively inexpensive since it is the type of system used as a standard throughout the television industry. Moreover, it has some very worthwhile advantages. For example, two different images may easily be combined electrically if their video is in synchronization, and this is impossible with random X-Y-Z deflection. As will be discussed later, the ability to mix images is one of the simpler operations which can be performed with off-the-shelf equipment made for the television industry.

Other video systems exist but are highly specialized and lack the support of as many product lines as enjoyed by X-Y-Z and TV. For this reason they are excluded from this discussion.

It is certainly true that a raster-scan system places more rigid restrictions on modules providing it with input than does an X-Y-Z system. Whether these restrictions are good or bad for a system taken as a whole depends on the

nature of the modules generating the deflection input and on the nature of the data base represented by the images. If the data base is a scene and the module is a TV camera, video is obviously highly convenient. However, most real-world situations lack such obvious solutions, and often one is forced to carefully consider his data base and then either find or construct a device for transforming it into one of the deflection inputs. Many such devices exist and these devices employ many radically divergent techniques for performing the transformations. It is, furthermore, difficult in the case of digital data bases to say what their "natural form" is. If one could do this, perhaps he could characterize an ideal deflection-generating device. In practice, the form of the data base is instead tailored to suit the available devices.

The deflection system most commonly used for driving storage CRTs is the X-Y-Z organization. It is simple and relatively easy to use since the image needs to be drawn only once between erasures. Other systems use either X-Y-Z or TV-raster video deflection as seems most suitable. While suitability largely depends on the modules which generate the deflection signals, there is one general statement which can be made. For displays requiring crisp lines, small-size pictorial elements, or extreme precision in the placement of points, lines, or other elements, X-Y-Z deflection offers the greater capability. For displays requiring extreme *quantity* of pictorial elements, video is preferable since it requires the same information *bandwidth* for any display *complexity*, while X-Y-Z requires that bandwidth increase proportional to complexity. If the application requires mixing, recording, or transmission of images, video is also indicated.

ELEMENTS PRODUCING AND PROCESSING X-Y-Z SIGNALS

Elements Producing X-Y-Z Signals

For display of digital data, the fundamental device is the digital-to-analog converter (DAC). Three of these transform any arbitrary source of ordered triads of digital values into X-Y-Z deflection signals. Fairly simple devices built around DACs are frequently used to connect digital computers to storage CRTs. Since each triad submitted to the DACs defines only one combination of position and intensity, and since their output values change in a discontinuous fashion, these devices must usually create displays as a succession of dots or minute lines. To draw a line, they must be "stepped" along the line incrementally. A single line may require hundreds or thousands of steps, from which it should be apparent that a DAC will require vast quantities of digital input for a complex display. This translates into an extreme memory or computing load for the module providing the DAC with input. Most systems do not attempt to store the digital input for each minute step of a line and instead contain some element which calculates the direction of each step, computes its coordinates, and passes these coordinate values to the DACs as needed. The amount of time required for a general-purpose computer to perform these calculations is too great to permit rapid refreshment of complex images on non-storage CRT displays, so such an approach is seldom taken. It is, however, viable to drive a storage CRT in this fashion.

For refreshed (i.e., non-storage) displays, there exists a group of devices called *random-vector generators*. These devices convert digital *descriptions* of points, lines, symbols, and, in some cases, curves into appropriate X-Y-Z deflection signals. By accepting descriptions, they require far less input than do simple DAC modules. While vector generators differ somewhat in resolution and electrical noise in the output signals, the greatest spread in their capabilities occurs

in the *speed* with which these pictorial elements are generated. This is an important performance area since for each CRT phosphor there is a minimum frequency of display refreshment below which humans will perceive flickering. Thus, a given vector generator's speed determines the maximum complexity beyond which it cannot produce a flicker-free display on a given CRT. While speed can be limited by the CRT's deflection system, the state of the art in refreshable complexity is currently limited by vector generators rather than by CRTs.

The best vector generators act as small computers, directly connected to memories containing descriptions of graphic elements which the devices repetitively fetch and interpret. Often the memory involved is shared with a general-purpose computer which may alter the descriptions at will.

This last organization is suitable for extremely dynamic displays. Very little work is required to alter the display description in the memory, and the alteration is implemented pictorially the next time the vector generator comes by the altered area. Vector generators are widely used in command and control systems.

X-Y-Z Processing

The only means of combining two sources of X-Y-Z deflection input is by time multiplexing, since the CRT beam can only be in one place at a time. Moreover, this sort of multiplexing cannot be accomplished without the cooperation of each of the multiplexed sources. The recording of such signals is seldom done due to their bandwidth. However, there are some processing functions which are much more simply done with X-Y-Z signals than with video.

A display may be translated or changed in scale (zoom) by simple linear transformations. Rotation is feasible if the vector generator or DAC system involved is relatively slow. There even exists a product line which includes modules capable of performing the necessary analog calculations to transform

three-dimensional coordinate signals into X and Y, complete with perspective movable by other analog signals. Many of these capabilities have found their way into working systems.

ELEMENTS PRODUCING AND PROCESSING RASTER-SCAN VIDEO

Elements Producing Raster-Scan Video

There is a wide variety of devices which generate video signals. Television cameras produce video directly from a visual image. Video tape recorders are capable of storing video displays regardless of their source and playing them back at a later time unchanged. However, the devices of greatest interest to us in considerations of MSRF are those which accept digital input and produce raster-scan video images therefrom. There are three general classes of devices capable of performing this function.

Digital image memories. The most commonly used technique for producing video signals from digital input is also the least elegant, with the most "brute force." The procedure is to build a memory which has an exact one-to-one correspondence with each point on a rectangular display. Each element of this memory represents the brightness of the corresponding point on the display surface; if a simple black and white display is desired, all that is required is one bit of memory for each point. If gray scale information is to be presented, then each element must consist of several bits. In operation, the contents of the digital memory are clocked out and converted to Z-axis or video information in synchrony with the scan taking place on the television system. This is a technologically simple process, and is conceptually very similar to the situation one has with the previously discussed plasma panels. In fact, most of the software problems which result in the case of the plasma panels apply equally to digital image memory display systems. It is very difficult to solve the problem of dynamic vector graphic displays in the general case with this sort of device. On the other hand, since the output

of the devices is in the form of video, mixing can be performed and much flexibility results. As far as information density is concerned, a full digital image memory will produce the greatest possible information density on a CRT display since, in the case of a 1,000 line display, one million points may be discretely controlled.

On-the-fly raster generators. A fairly recent and somewhat exciting device made by several manufacturers is capable of producing raster-scan video information from digital input in a much more flexible fashion than is possible with full-image memories. The characteristics of this sort of device combine the logical simplicity of X-Y-Z deflection and refreshed CRT displays with the signal manipulative convenience of raster-scan systems. Like the random-vector display systems, these operate from a set of descriptions of graphic elements to be displayed. Although the algorithms and hardware used are proprietary, what takes place is essentially that a very fast, very smart processor follows the scan down the screen for each frame, supplying that information which is relevant to each scan line of the display as it is being output. This requires extremely rapid referencing of the data base which describes the display to be produced and, while is degenerates severely in the case of extremely complex displays, dynamic effects can be produced with extreme logical simplicity in the input. After some consideration it has become apparent to us that by combining a full image memory with an on-the-fly raster generator, one can overcome most of the deficiencies of the two systems taken individually. That is to say, high density but relatively slow-changing information such as processed passive sonar displays can be easily produced in a full image memory, while highly dynamic graphic elements such as symbols and cursors may be produced separately via the on-the-fly technique without disturbing the extremely dense display elements which would be otherwise difficult to reproduce rapidly. The video outputs of the image memory and

on-the-fly generators could then be easily mixed to produce the desired composite image on a CRT display. The implications of such a combination will be discussed at length later.

Scan converters. Scan converters are very specialized cathode-ray tubes which are capable of converting displays produced by almost any CRT deflection system into raster-scan video. This is achieved by using one writing beam to place information via the arbitrary deflection system onto a phosphor screen. The other end of the tube removes this information from the screen by scanning it in a traditional video format. Superficially, this would seem to be a very effective means for obtaining video output from random-vector input. However, the tubes are somewhat difficult to use, tend to require frequent adjustment, and are very expensive. Furthermore, the resolution of graphics produced by such devices will always, for a given scanning resolution, be worse than that producible by a digital television technique using the same display resolution. The reason for this is that in the case of the image memory and on-the-fly graphic generation techniques, graphic elements may be matched precisely to the field of available pixels and extremely small graphic elements may be produced with clarity by forming them from dot matrices. However, one does not have such precise control in the case of scan conversion; while an image memory can produce a point which occupies precisely one pixel on the video display, a scan converter has a very small probability of seeing the same point on only one of its scan lines. Consequently, most graphic elements will appear with approximately half the resolution or half the clarity on a scan-converted display as they would have exhibited if they were produced by an image memory device. We have viewed scan conversions of various graphic displays with disappointing results.

Raster-Scan Video Processing

Conveniently, the television industry has generated numerous devices capable of transforming or otherwise processing

video information. Some of these may be of interest for MSRF. Video mixers enable the combining of two or more video displays without imposing any special requirements upon the devices generating those displays. Various parameters of the mixing operation can be controlled, such as the relative brightness of the displays being mixed. Another device of interest is the complex function-generating mixer. This device enables the selective combination of several images, and includes the capability to perform such compositional functions as to translate or mask individual inputs.

TIME-MULTIPLEXED RANDOM-VECTOR/RASTER-SCAN SYSTEMS

One means of employing the advantages of both random-vector and raster-scan techniques in a single display system is to time-multiplex display generators of each type. A raster-scan system methodically sweeps the electron beam across the display screen at a fixed rate, during which time the beam is modulated by an appropriate video signal to produce a single scan line. At the end of a scan line, the system enters a "retrace" phase, during which the beam is blanked (i.e., reduced in intensity so that it makes no visible trace) and repositioned to the beginning of the next scan line to be drawn. It is possible by appropriate control logic to connect a random-vector generator to the CRT deflection system during this retrace time to permit lines, symbols, or characters to be drawn at arbitrary positions on the display surface. Thus, for example, the high-density, less dynamic information typical of passive sonar data could be presented from an image memory on the display by the raster-scan method, while low-density, highly dynamic information such as movable cursors, tactical symbols, and alphanumeric target information read-outs were drawn during retrace intervals by the random-vector generator. This method is practical, and is currently used by the Hughes UYQ-21 acoustic display console and by certain Motorola military display systems.

However, the time-multiplexed approach has two important limitations of particular significance for MSRF. First, there is a fairly severe constraint upon the amount of random-vector drawing that can be done, because the raster-scan re-trace intervals must be brief, to maintain the raster-scan refresh rate high enough to avoid intolerable display flicker. It is true that this is a limitation of actual state-of-the-art military systems (e.g., the UYQ-21), but it is desirable to have a *research* display system that can go beyond the capabilities of operational (or nearly operational) Navy display systems wherever possible.

The second important limitation of the time-multiplexed approach as regards MSRF is economic. Currently, the only systems we know of which employ this technique are the Hughes and Motorola military systems. Because they are militarized systems, and in various stages of pre-production, their costs appear to be prohibitive as far as MSRF is concerned.

MSRF Console Displays

DISPLAY DEVICE SUITABILITY FOR MSRF

In order to determine the suitability for MSRF of the various display devices afforded by the technology reviewed above, the display *requirements* of MSRF must be compared to the characteristics of each of the potential display devices. Our analysis of Navy systems likely to be of interest in MSRF-supported research indicates that there are five general classes of data which the MSRF console display systems may be required to present. Each of these will be discussed in turn.

Alphanumerics

Alphanumerics are symbols consisting of the letters of the alphabet, the numerals 0 through 9, and the common punctuation and other marks. The symbols commonly considered to

be "alphanumeric" may be regarded as those defined by the current USA Standard Code for Information Interchange (ASCII). Some systems use only a subset of the entire ASCII set which excludes lower case alphabet characters and certain punctuation and other marks, while other systems may use a superset of the ASCII set which includes symbols for special purposes. Whatever set of alphanumerics a system is provided with, its symbols are usually presumed to be employed fairly often, and therefore the instructions for drawing them on the display screen are usually hard-coded in the system, for example, by the use of read-only memories. With the advent of generalized computer-controlled displays in Navy systems, alphanumeric information previously presented directly on mechanical or electronic discrete readouts, inscribed legends, etc., have come to be displayed graphically. An alphanumeric capability is fundamental to many Naval systems of interest and, therefore, the capability is fundamental to MSRF.

Lines and Symbols

One of the most important differences between general-purpose computer-driven displays and older displays dedicated to the presentation of sensor data is the ability of the former to draw lines and symbols (i.e., symbols in addition to those which are part of the alphanumeric set discussed above). The ability to draw lines of random lengths and positions provides a graphic display system with the capability of presenting such useful information formats as geographic situation summaries, wherein a geographic coordinate system may be detailed together with symbols indicating the positions and types of various tactical units, and lines (dotted, dashed, solid, etc.) indicating past paths or future predicted paths, and so on. Special symbols (which are not part of the alphanumeric set and are therefore not part of the system firm ware) may be composed by software from appropriate combinations of lines.

Radar Data

Most of the radar sensor data presented in systems of interest for the MSRF design are presented in a plan position indicator (PPI) format. Since radar sensor data are typically derived from a rotating radar antenna azimuthally scanning the area under surveillance, it is natural to consider the PPI in terms of a polar coordinate system. The origin of the polar coordinate system is the location of the radar antenna, which is usually (but not always) depicted in the center of the PPI presentation. Since the speed of radar wave propagation to and from targets is far greater than the rate of radar antenna rotation, the radar PPI displays at any given moment the radar reflectivities of all targets at a given bearing out to the limit of the range scale being used. These radar reflectivities are indicated on the PPI by intensity modulation, and the "sweep line" thus formed extends from the origin radially outward to the maximum range scale. This "sweep line" is synchronized with radar antenna rotation so that one has the familiar radar PPI presentation of the sweep line scanning around the tube, "painting" radar images on the PPI. The CRT phosphor of a radar PPI tube is typically chosen to have a persistence approximately equal to the time taken for the radar antenna to complete one revolution, so that the radar image from the previous sweep is quite faded (but still apparent) when a new sweep passes a given bearing. These characteristics have important implications for display design, as will be discussed presently.

Sonar Data

PPI. Active scanning sonars have traditionally used the PPI format. The polar coordinate system is a natural way to view the sonar PPI, for the reason discussed pertaining to radar PPI presentations. However, there is a very significant and apparent difference between radar and sonar PPI presentations. This difference arises because the speed of sound in

water is very much slower than the speed of radar waves in air. Consequently, the sonar azimuthal scanning rate is much greater than the rate at which echoes from all targets on a given bearing are returned to the sonar. When a radar transmits on a given bearing, echoes from all targets out to the maximum range are returned in a matter of microseconds, and they are therefore presented on the PPI format "simultaneously," as far as the human eye is concerned. On the other hand, when a sonar transmits on a given bearing, it may take many seconds for echoes at the maximum range to be returned, and it is impractical to stop the scanning process on a given bearing to wait for all possible echoes. Consequently, the sonar scan continues, and as that scan passes a given bearing, the intensity modulation presented on the sonar PPI corresponds to echoes (or lack of echoes) in a very limited range band, which is visually indicated on the sonar PPI as a spot. As the sonar scan continues, this spot is moved around the tube in an angular direction in synchrony with the sonar scanner, and outward in the radial direction to correspond to the range band covered at each moment as time passes since the last sonar transmission. This scanning method gives rise to the typical "expanding spiral scan" of the sonar PPI. The persistence of the CRT phosphor is usually selected to be approximately equal to the time taken to achieve complete coverage of the area under surveillance, so that the sonar "picture" from a previous transmission is fading away (but still apparent) at the time it is "repainted" by the succeeding scan.

A-scan and B-scan. Active sonars have more recently employed preformed beam receivers in addition to the scanning receiver which gives rise to the sonar PPI presentation. Preformed beam receivers "listen" in one direction only and this gives them some considerable signal processing advantages over the scanning receiver, which only periodically "listens" on a given bearing as the scan passes by. It is not relevant to discuss these signal processing advantages here; however, the

display consequences resulting from the use of preformed beam receivers are relevant and considerable. The outputs of preformed beam receivers are generally presented in A-scan or B-scan formats. In these formats, the output of each preformed beam receiver is presented by an oscilloscope trace which is linearly deflected in proportion to time since the last transmission on that bearing (thus, the deflection corresponds to range of target echoes), and target echoes modulate the beam by either deflection modulation (A-scan) or intensity modulation (B-scan). (Some "A-scan" systems actually use a combination of deflection and intensity modulation.) Generally, the orientation of the trace is such that the spot of the electron beam defining the trace moves upward on the display as time passes. Thus, the vertical dimension represents range, with the bottom of the tube corresponding to the shortest range displayed and the top of the tube corresponding to the longest range displayed. There are always a number of preformed beam receivers (typically, anywhere from 10 to 40), each corresponding to a different bearing sector, and the traces corresponding to the outputs of these various receivers are generally displayed side by side, so that the typical format is one of range (the vertical dimension) versus bearing (the horizontal dimension). Most preformed beam systems also incorporate some form of memory, so that the traces of some number of preceding transmissions are kept in analog or digital form and are displayed on the CRT side by side. The long-persistence phosphors typical of radar and sonar PPI displays are not used with the A-scan and B-scan formats, because they are typically presented on refreshed raster-scan display systems.

Passive sonar. All passive sonar systems likely to be of interest in the design of MSRF are of the preformed beam type discussed above with respect to active sonar. However, passive preformed beam receivers "listen" for noise generated by targets on each bearing covered by the system, as opposed to listening for echoes resulting from an active transmission

from ownship. Broadband passive sonar, as the name implies, "listens" on each bearing covered across a relatively broad acoustic spectrum, and the information is typically displayed on a bearing-versus-time format, where bearing is usually displayed on the horizontal dimension and time on the vertical dimension. With a preformed system, the output of a given receiver ("listening" on a given bearing) gives rise to one of the several vertical lines apparent on the display, each corresponding to a receiver associated with a given bearing. The output of narrowband processing is usually displayed in a frequency-versus-time format, which nonetheless has a very similar appearance to a broadband sonar display (the interpretation of the information in the display is, of course, quite a different matter). Since the narrowband processor display usually pertains to only a given bearing, and there may be very many bearings covered by a system, there may be a need to view narrowband formats individually for each of several bearings; or the narrowband formats may be "condensed" or compressed in the vertical (time) dimension by various means so that a number of narrowband processor displays, each corresponding to a different bearing, may be viewed on a single CRT display format. All types of passive sonar displays are characterized by a very great information density which changes relatively slowly. These characteristics have important implications for the selection of a suitable display system, as will be discussed shortly.

Imagery

Imagery, as we are concerned with it, will be encoded and displayed in conventional or high resolution television picture (i.e., raster-scan) formats. Imagery can be important in such Naval systems as remotely piloted vehicles or in remote underwater manipulative tasks. Although imagery might be of secondary concern to the research applications of the MSRF, the raster-scan technology underlying these imagery techniques is compatible with the raster-scan technology used

for some of the other data presentations, and thus it may be possible to include the capability of presenting and dealing with imagery in the MSRF with very little added investment.

Display Device Suitability for Each Class of Data

Each of the five classes of data discussed above presents somewhat different requirements for a display system. In Table 7 we summarize the suitability of various display devices for presenting these five classes of data. The five classes of data are represented in the columns of Table 7, and each major column is subdivided into three parts, representing the dynamic nature of the information presented in each class. Each class of data may be presented in an essentially static (S) mode, requiring very infrequent changes. However, each class of data may require a low-level dynamic capability (DL), requiring updating at more frequent intervals; or it may require a high level of dynamic capability (DM--maximum dynamic capability for a particular class of data), requiring relatively rapid changes in the displayed data.

The rows of Table 7 represent the various devices previously discussed under Display Technology which should be considered for application in MSRF. Entries have been made in the table at the intersections of each row and column to *roughly* summarize the suitability of each display device for each particular data type. No entry in a cell indicates that the device is thoroughly suitable; the letter R indicates the device is suitable with some reservations; the letters SR indicate that the device may be suitable, with severe reservations; and the shaded cells indicate that the device is unsuitable for displaying a particular type of data.

Let us consider the first row, having to do with the suitability of storage CRTs for displaying various data types. For alphanumerics, the storage CRT is thoroughly suitable for displaying information that does not need to be changed selectively or frequently. For a low-level dynamic requirement,

TABLE 7
SUMMARY OF DISPLAY DEVICE SUITABILITY FOR EACH CLASS OF DATA

DISPLAY DEVICE	A ALPHANUMERICS			B LINE SYMBOLS			C RADAR DATA			D SONAR DATA			E IMAGERY		
	S	DL	DM	S	DL	DM	S	DL	DM	S	DL	DM	S	DL	DM
STORAGE CRT		SR			SR		SR	SR		SR	SR		SR	SR	
PLASMA PANEL					SR	SR	SR	SR	SR	SR	SR	SR	SR	SR	
VECTOR CRT															
RASTER CRT:															
IMAGE MEMORY					SR	SR	SR	SR	SR			R			SR
SCAN CONVERSION						SR			SR			SR			SR
ON-THE-FLY															
TIME-MUX CRT															

THOROUGHLY SUITABLE

R SUITABLE WITH RESERVATIONS

SR SUITABLE WITH SEVERE RESERVATIONS

UNSUITABLE

the storage CRT is suitable with severe reservations; the reservations being that data may be *added* to the display but data may not be *selectively* erased to be replaced by other data. Replacing previously written data requires erasing the entire screen, a process which takes approximately 500 milliseconds and produces a relatively bright flash, owing to the erase-flooding of the entire screen. The storage CRT is unsuitable for highly dynamic alphanumerics because of this reason. The same set of considerations apply to displaying lines and symbols with varying degrees of dynamics on the storage CRT. For radar data, the storage CRT may be suitable, with severe reservations, for static or slowly changing data. A display can be built up on a storage CRT in a scanning pattern analogous to that of the radar PPI. However, the storage CRT is essentially a bistable device, incapable of producing shades of grey (between fully "off" and fully "on"); and furthermore, the entire display must be erased to begin a new display, so that the observer does not have the "after image" left behind on a standard radar PPI with which new information may be correlated. Because of these limitations, we feel that the storage CRT is totally unsuitable for highly dynamic radar presentations. All these considerations mentioned with respect to radar data also apply to the use of storage CRTs for the presentation of sonar data, and an additional reservation is brought into play because of the typically extreme density of passive sonar data. Most storage CRTs have an upper limit to the area of the storage screen that may be placed in the "on" state, and this restriction may make it impossible to present passive sonar data in any realistic manner. The limitations of the storage CRT with respect to lack of grey scale and limitations on the total amount of available area that may be illuminated make it marginally usable for the presentation of any sort of imagery information; certainly, it is thoroughly unsuitable for any sort of dynamic images because of the requirement to totally

erase the screen to update displayed information. Thus, moving parts of a scene (or symbols or any other image component that is to be perceived as moving in some manner across the face of the tube) simply cannot be displayed on a storage CRT device.

In the second row, the suitability of plasma panel display devices is considered. It can be seen from the table that the plasma panel is suitable for all types of alphanumerics. The plasma panel *does* have selective erase capability, so that in conjunction with fast character generation capability, the plasma panel can satisfy the needs for presenting alphanumeric information. However, with respect to lines and symbols, one must consider the plasma panel with severe reservations for very dynamic displays because moving parts of a line drawing on the plasma panel requires a point-by-point erasure of the display element to be moved, including computation of points of intersection with stationary parts of the display so as not to leave "gaps" in those stationary parts, and a point-by-point redrawing of the moving component elsewhere. This requirement places an extreme load upon the processing capability of the device driving the plasma panel. Even in static displays, the quantization of the display into pixels results in a degradation of line and symbol quality, since straight lines (except at ideal angles) and smooth curves cannot be drawn without discontinuities. For most applications though, this is more a consideration of aesthetics than of functional unsuitability. The plasma panel could be considered suitable only with severe reservations for the presentation of radar, sonar, or imagery data because of its lack of grey scale and because of the computational load involved in presenting such displays. It must be regarded as thoroughly unsuitable for highly dynamic imagery presentations.

In the third row, the suitability of random-vector CRT display systems is considered. It can be seen that the vector CRT is thoroughly suitable for alphanumerics and lines and

symbols. Refreshed random-vector display processor technology is sufficiently advanced that rather complex alphanumeric and graphic information may be produced at a sufficient rate to produce reasonably complex displays without intolerable flicker. These displays are especially suited to drawing linear information, because they have no inherent "pixel" structure on the display screen; any portion of the display screen may be illuminated by the display processor, and a straight line from "A" to "B" is drawn directly as a straight line between those two points, and appears as a straight line to the eye (assuming that certain inherent distortions in the deflection system are suitably compensated for). However, the vector CRT must be regarded as unsuitable for radar, sonar, or imagery data because the density and complexity of displayed data in these classes exceeds the capabilities of random-graphic display processor technology to regenerate the entire display fast enough to avoid flicker. Further, few (if any) vector CRT systems provide for intensity modulation beyond one or two steps, and use of this capability in most available systems aggravates their flicker problem far further.

In the fourth row, the suitability of raster-scan CRTs is summarized. CRTs which employ raster-scan techniques are generally capable of doing a fairly good job at displaying any of the classes of data of interest to us. There are a few implicit reservations; for one, raster scanning implicitly quantizes the display system in one dimension (normally the vertical) since the display is composed of a finite number of scan lines at fixed positions. Regardless of the method used for generating the raster, this quantization makes it difficult to represent lines or line segments whose slopes are nearly horizontal. Another slight reservation occurs in the case of radar PPI data, since there is no way to reproduce the full resolving capability of modern radar displays by horizontal raster techniques. We feel, however, that this

loss of resolution is insignificant for the purposes of MSRF. Beyond the general suitability of raster CRTs, there are distinct strengths and weaknesses pertaining to each method of generating video; these will be discussed next, and are summarized in rows 4, 6, and 7 of Table 7.

The suitability of image memories for the various classes of data is similar to that of the plasma panel in that the two devices are geometrically and logically analogous. Differences in the assessments result from the capability of image memories to show each pixel as one of a fixed set of grey shades (normally some power of 2). Image memories have been built which can represent 256 shades of grey, although 8 is a more typical number. The devices are suitable for alphanumeric data, with no reservations. Their qualities for lines and symbols are the same as those of plasma panels, suffering from display quantization and from the processing effort required to update displays in areas where multiple picture elements share pixels. For radar data, severe reservations are maintained because persistence effects cannot be reproduced without special CRT phosphors, although if this path were taken the rating would improve to "R" for all but the highest updating speeds; the "R" remains because of the processing load involved in continually outputting graphic data to the image memory. An "R" is likewise appropriate if the radar display to be simulated is itself implemented by an image memory (as is in fact planned for PPI presentations on the UYQ-21 in the SSSMP). For sonar applications (other than PPI), image memories are suitable since this is the very technique used in the majority of modern sonar consoles. However, at high updating speeds, commercial image memories fall somewhat short; the most common updating mode of existing sonar displays, known as "waterfalling," is implemented by special hardware in the image memory. To achieve the same effect with a conventional image memory, each pixel in the display must be rewritten. This

represents a sizable data transfer and results in substantial computer loading even at modest updating speeds. The suitability of image memories for imagery is excellent, except at high updating speeds for which the devices are generally not designed. A 512 x 512 point memory, altered at 30 Hz, implies a transfer rate of 7.8 million pixels per second, or 127 nanoseconds per pixel. Minicomputers lack the mass storage capacity and speed to continually update such displays. (The largest mass storage device available from DEC could only hold 50 seconds' image data, under the conditions described above, for a 16-grey-level display, and could not transfer the data fast enough for even a 6 Hz updating frequency. The mass storage systems for even extremely large computers are not much better.)

Scan conversion is generally suitable for all classes of data, so long as displays are static or are infrequently updated. At low updating frequencies, single-ended scan converters may be used; these devices' capabilities in this mode are analogous to those of image memories, although different adjustments are required for some of the classes of data, precluding many combinations. At high updating frequencies, it is necessary to employ double-ended scan converters, whose characteristics are *not* comparable to those of image memories. Single-ended converters use the same deflection systems, D/A converters, and other analog components for both reading and writing. As a result, it is easy to "find" the same point on the phosphor screen twice. Double-ended tubes employ *separate* deflection systems, and it is (according to engineers who work with them) almost impossible to adjust both ends of the system to track each other within the spot size of the system, or to keep those adjustments stable. Substantial losses of resolution result, causing our severe reservations.

On-the-fly video generators are fairly specialized. They are functionally analogous to vector CRTs, implying that they

are thoroughly suitable for alphanumerics, lines, and symbols. On the other hand, they are unsuitable for extremely dense displays, which precludes their use for radar, sonar, or imagery data. Admittedly, they still produce lines quantized into pixels and they cannot produce displays quite as complex as the best vector CRT systems; nevertheless, these reservations are slight relative to most of the others we have mentioned.

In the last row of Table 7, the suitability of time-multiplexed CRT systems is summarized. In terms of military display requirements, time-multiplexed CRT systems must, by definition, be considered thoroughly suitable since this is the technology *used* in the most sophisticated military display consoles. Admittedly, there are isolated instances where other technologies are superior; yet, the capabilities of these systems tend to influence military display requirements, just as the systems may themselves be expected to evolve in cases where the requirements cannot be compromised. Until military display manufacturers evolve a superior technology, these systems must be the standards against which would-be simulators will be judged.

Display Device Suitability for Typical Combinations of Data Classes

We have discussed display device suitability with respect to each of the five classes of data considered individually. However, Navy systems typically require the display of several of these classes of data simultaneously. For instance, tactical data systems (TDSs) require the display of highly dynamic alphanumerics, highly dynamic lines and symbols, and moderately dynamic radar data in combination; modern passive sonar systems require the display of highly dynamic alphanumerics, highly dynamic lines and symbols, and moderately dynamic passive sonar data; and imagery systems of possible interest to MSRF might require the display of highly dynamic alphanumerics, highly dynamic lines and symbols, and highly dynamic imagery

data. The MSRF display system certainly should be able to simulate the first two of these example systems and possibly the third. Therefore, the display implications of these typical combinations of data classes must be considered.

A summary of display device suitability for these typical combinations of data classes is shown in Table 8. The cell entries in this table have the same meaning as those of Table 7, and they represent a logical combination of the cell entries in that table.

It can be seen from Table 8 that storage CRTs are not suitable for simulating any of the three example systems because of their unsuitability for displaying dynamic data. Plasma panel displays are suitable only with severe reservations for simulating TDS and sonar systems because of their previously stated limitations with respect to displaying radar and sonar data, and highly dynamic lines and symbols; they are not suited to highly dynamic or grey-scale imagery display. Vector CRT displays are not suitable for TDS, sonar, or imagery systems because of their physical inability to display any sort of video information.

The high-resolution raster-scan monitor is suitable for simulating all three systems, and is in fact the display device employed in many of these systems. With respect to video generation, it can be seen that digital image memories are suitable only with severe reservations for the three systems because of their previously mentioned line and symbol, radar, and imagery data limitations. Scan conversion video generation is suitable only with severe reservations because the highly dynamic requirements of the three example systems imply use of double-ended scan converters, with their previously discussed limitations. On-the-fly video generation is unsuitable for simulating any of the example systems because it cannot produce radar, sonar, or imagery data.

TABLE 8
SUMMARY OF DISPLAY DEVICE SUITABILITY
FOR TYPICAL COMBINATIONS OF DATA CLASSES

DISPLAY DEVICE		TDS DISPLAY	PASSIVE SONAR	IMAGERY
STORAGE CRT				
PLASMA PANEL		SR	SR	
VECTOR CRT				
RASTER CRT:				
	IMAGE MEMORY	SR	SR	R
	SCAN CONVERSION	SR	SR	SR
	ON-THE-FLY			
	TIME-MUX CRT			
RASTER CRT WITH:	ON-THE-FLY & IMAGE MEMORY	SR		R
RASTER CRT WITH:	ABOVE PLUS SCAN CONVERSION			R
RASTER CRT WITH:	ABOVE PLUS EXTERNAL VIDEO			

The time-multiplexed CRT technique is seen to be suitable for all applications. The excellence of this technique derives from its hybrid nature which combines random-vector graphic display generation (which is ideally suited to the production of highly dynamic alphanumerics, lines, and symbols) with either direct analog radar deflection generation for radar PPI data or digital memory video generation for raster-scan sonar data displays. The concept of *combining* various, complementary display generation techniques to achieve a system with few inherent limitations is an excellent one. Unfortunately, in the case of time-multiplexed CRTs, it is also an expensive one.

However, the concept of combining the outputs of display generators with complementary capabilities may be accomplished without expensive and complex time-multiplexing. The various raster-scan display generation techniques (digital image memory, scan conversion, on-the-fly, plus TV cameras and video tape recorders) can produce compatible raster-scan video signals which may be easily combined electronically using video mixers and other television industry devices, and the combined video signal may be viewed on any number of readily available (and inexpensive) CRT monitors of various sizes, shapes, and phosphor colors and persistencies.

The implications of this approach for MSRF are summarized in the bottom three rows of Table 8. When on-the-fly and digital image memory techniques are combined by video mixing, we see that the configuration is suitable without reservation for the simulation of passive sonar systems. This is so because the on-the-fly system is totally suited to supporting the high-speed alphanumerics and lines and symbols requirements, while the image memory is totally suited to the medium-speed sonar data requirement. The on-the-fly plus image memory system is also suitable, with some reservation, for simulating TDS systems because, with

long-persistence phosphor CRT monitors in conjunction with appropriate image memory manipulation, we believe a reasonable digital raster-scan simulation of conventional analog rotating-sweep radar PPI displays might be achieved. This same approach would apply to simulating analog spiral-scan sonar PPIs.

With the addition of a scan-converter module to the on-the-fly plus image memory configuration, the reservation concerning TDS systems is dropped. The scan converter would be driven by an analog rotating-sweep radar PPI simulator so that a very credible TDS simulation would result. (The scan converter could also be driven by a spiral-scan sonar PPI simulator to support research involving that particular display.)

Finally, in the last row of Table 8 it can be seen that with the addition of appropriate video from an external source (e.g., from a TV camera or video tape recorder), the reservation with respect to highly dynamic imagery system simulation is dropped, and we have a total display system capable of satisfactorily simulating TDS, sonar, and imagery systems.

The only other potential MSRF display system approach with totally adequate functional capabilities would involve the time-multiplexed technique. The choice among approaches must involve considerations of performance, cost, and reconfigurability which will be discussed next.

RECOMMENDED DISPLAY SYSTEM

The foregoing analysis effectively eliminates most display devices from further consideration. Indeed, only three survive, and of these only one appears to come sufficiently close to meeting the requirements and constraints of the MSRF that it deserves recommendation. The trade-off considerations among these three systems are discussed below, and the recommended system is described in detail.

Time-Multiplexed CRT Systems

Given an unlimited budget, it would be tempting to choose exact duplicates of the very display systems it is desired to study. It would certainly seem that such a path would lead to rapid and straightforward implementation of displays for each experiment. Unquestionably, the resulting display fidelity would be superb. We did, in fact, explore this possibility with dismaying results. Taking the Hughes UYQ-21 console, for example, unit prices range from $250,000 to $450,000. (The latter price is for a sonar configuration.) Additional costs would be necessitated to provide water cooling and, for TDS use, additional hardware would be required. The consoles are not themselves configurable except within the constraints of their inherent shapes, and they certainly are not as flexible as desired for the MSRF. Furthermore, delivery times are on the order of 1 year.

Quite evidently the apparent advantages of using these devices vanish in light of these considerations and of the objectives and constraints of MSRF.

Plasma Panels

These devices survive as alternatives only because of their modest price--around $5,000 to $10,000 per console. As has been noted, their suitability for TDS and sonar displays is conceded only with severe reservations in areas of display fidelity and dynamic effects. It should be noted that plasma panels, such as those used in PLATO terminals, have been used by NPRDC personnel to simulate ATDS console functions for training purposes; this is a laudable action since the plasma technology is economically suited to training environments which incorporate large numbers of consoles.

However, while their acquisition costs are attractive, their general unsuitability leads to concealed recurring costs. These result from the ingenuity, in both experimental

and software design, required to overcome the weaknesses of the devices. While perhaps justifiable for training purposes where course software has a long productive life span, these costs detract significantly from the flexibility necessary in a research environment. Moreover, some of the areas of unsuitability (such as the inability of currently available panels to produce shades of grey) result in only the most superficial degrees of fidelity.

The difficulty of using plasma panels, coupled with their generally poor fidelity in simulating shipboard displays, leads us to conclude that they are inappropriate in view of the ambitious objectives of the MSRF.

Raster-Scan CRT Systems

A most satisfying compromise between price, fidelity, and ease of use has emerged from the study of raster-scan video technology. While no currently available *single* video source does a particularly good job of satisfying MSRF display requirements, one is happily not forced to employ single sources in building a system. As we have seen, the unique ability of video techniques to *combine* video signals from arbitrary sources enables us to employ separate generators whose strengths complement one another's weaknesses. The greatest single synergic payoff results from combining image memory and on-the-fly techniques. The essential components required for such a combination are separate on-the-fly and image memory devices, and a video mixer. Pictures from the two devices are combined by summing their brightness values at each point, producing a combined video signal. Adjustments can be made to vary the effects and appearance of this summation. Further, there is no practical limit to the number of independent video sources which may be combined, although the signal-to-noise ratio of the resulting display will usually suffer a slight degradation with each additional input.

Considerable flexibility is offered by such a system. The implementor of a given display is free to assign functions to those devices best suited for supporting them; examples of this flexibility will be presented shortly. The major technical difficulties involved in implementing such a system are similar to those which must be dealt with in a TV studio. In order to combine two video signals, they must be synchronized in time. Any error in *time* synchronization of the signals results in a corresponding error in *spatial* registration of the displays. Since this is a common problem, most devices which generate video signals are capable of operating in a mode such that the timing of their outputs is controlled by a pulse train received from an external source. All of the devices in the studio are synchronized by distributing the pulse train from a common *sync generator* to each, and this same solution may be employed in the MSRF.

A technical problem which is *not* so common in commercial television is that digital video generators do not all format their displays in precisely the same fashion. There are several "standard" numbers of horizontal scan lines subdividing a picture; one group is clustered around 525 lines and another around 1080. Further, some systems, in concert with similarly configured video monitors, paint a frame on the screen by tracing each scan line in succession from top to bottom, while others paint a frame by making two successive vertical passes, one of which traces odd-numbered lines and the other even-numbered lines (interlacing). It is difficult to combine the outputs of incompatible generators; indeed, one major commercial application of scan converters is the solution of such problems, although this method usually results in degradation of picture quality. To avoid this problem, we recommend that the MSRF select a single format and avoid procuring incompatible equipment. We recommend that this format be on the order of 1,080 lines to provide

the necessary resolution for relatively clean graphic lines and symbols, and for dense sensor displays.

There are several established manufacturers of commercial image memory systems; Hazeltine, Genisco, RAMTEK, Hughes, and Evans & Sutherland were the vendors who responded to our inquiries. On-the-fly devices are somewhat of a novelty, and, indeed, the only vendor among these who informed us of a proven capability to produce such devices was Hazeltine. Their commercial system is an adaptation of hardware used for the USAF AWACS program and for the USN Collision Avoidance System in the Integrated Bridge designed by HFR. Hughes informed us that they have such a device in the preliminary planning stages, although it is expected to be rather expensive. It seems likely that some of the other vendors mentioned above may develop their own versions of this sort of device, but at present it appears that Hazeltine is the only commercial source. We therefore spent some time in discussions with Hazeltine to tentatively configure a system and estimate its price.

The Recommended System

The present state of this concept is depicted schematically in Figure 6. Boxes drawn in solid lines represent the elements of the recommended configuration, which would provide the capability to simulate three NTDS or single-display sonar consoles. Boxes drawn with dashed lines are optional. Radar simulators, by means of scan conversion, would probably be required to simulate rapid-sweep PPI displays. TV cameras and/or video tape recorders would provide rapidly changing imagery. The remaining optional modules would enable the system to support three dual-display sonar consoles or up to six NTDS-like consoles.

The displays would be physically produced by 1,081-line high resolution video monitors, running at a 30 Hz frame

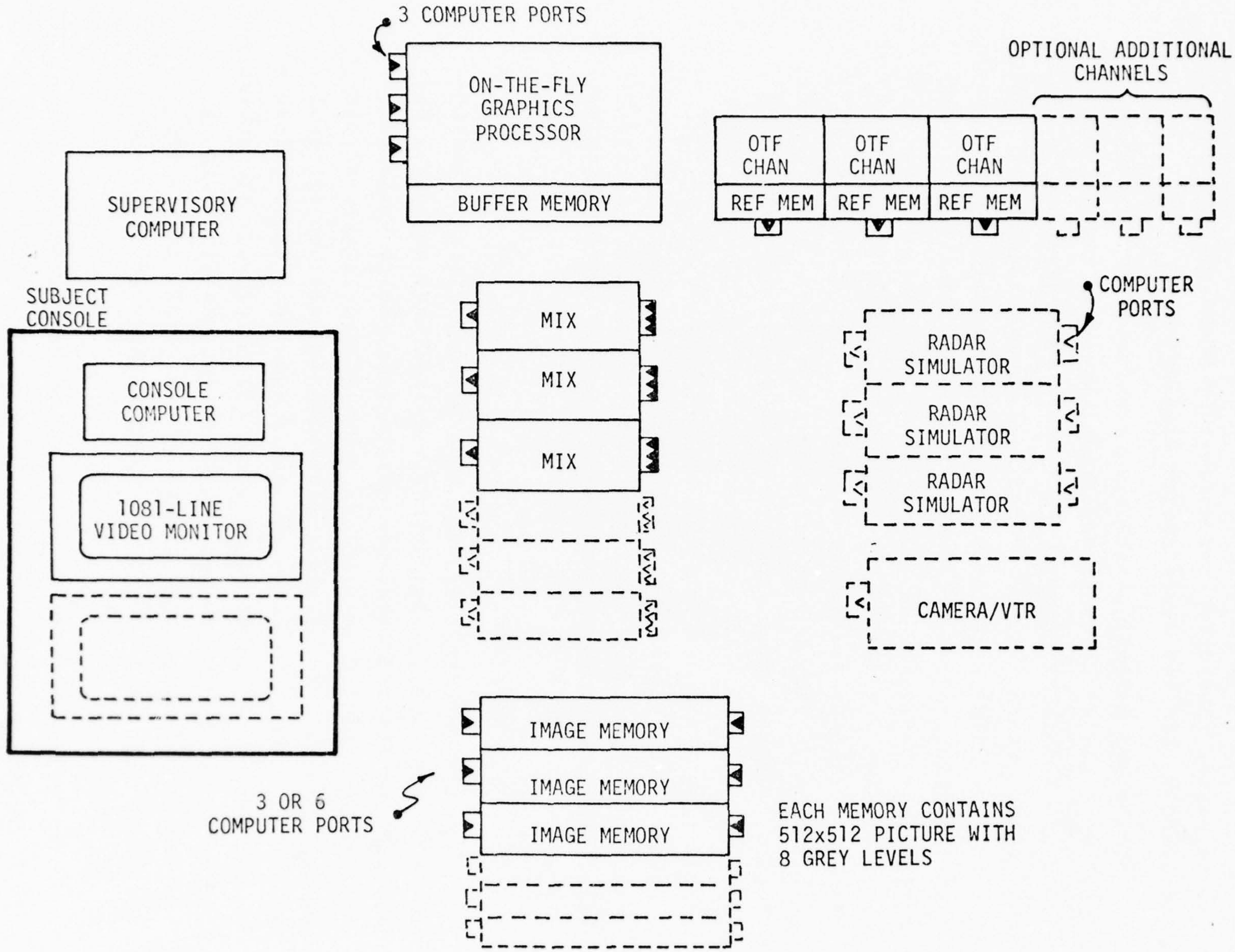


Figure 6. The recommended display system configuration.

refresh rate interlaced 2:1. Such monitors are available for approximately $1,500 to $2,000 depending on size, and standard units can be obtained with a wide variety of phosphors (P1, P11, P16, P19, P26, P31, P39, P40, P41, and P923 from one manufacturer).

The 1,018-line format is defined by the most critical element in the system, the on-the-fly generator. As made by Hazeltine, this is a modular device consisting of a graphics processor and one or more on-the-fly (OTF) channels (one per independent display). The processor receives commands from the computers and maintains an internal data base describing the contents of *all* displays. Whenever the computer changes this data base, the graphics processor determines what effect this change has on each independent display and updates the contents of the refresh memories associated with each display's OTF channel. Each OTF channel continually performs the operations necessary to transform the items in its refresh memory (descriptions of points, lines, and alphanumeric characters) into raster-scan video with an addressable resolution of 1,024 x 1,024 pixels. Each graphic element may be drawn at one of three shades of grey and may blink; individual line segments may be solid, dashed, dotted, or composed of alternate dots and dashes. One option for their device enables *selective display* of various classes of items in the graphics processor's data base; this would facilitate rapid display mode switching and other potential functions. The data base buffer and refresh memory sizes most recently postulated by Hazeltine would apparently support three independent displays on the order of complexity implied by concurrent display of:

- 2,000 connected lines
- 300 randomly placed tags of four alphanumeric characters
- 500 single characters at random locations

- 400 characters of text in 10 rows of 40 characters

In addition, three image memories are postulated. Each memory would contain 512 x 512 pixels, and each pixel could be displayed at any one of eight shades of grey. Special hardware would enable the image memories to produce 1,081-line output compatible with that of the OTF channels, although the resolution of the image memory displays would remain 512 x 512.

A common sync generator would drive all modules. Three video mixers would be provided; each would enable video from *any* three sources to be combined into a single signal. Thus, by cascading mixers, *all six* independent images produced by the system could be combined into a single display if desired. More practically, each mixer would normally drive one console display; two of the inputs would enable mixing of data from one each of the OTF channels and image memories, while the third would be available for inclusion of images from an external source such as a radar simulator or high-resolution television equipment.

These modules would be interconnected via a patch bay to facilitate the configuration of the system for various types of simulations. Another dimension of flexibility may be achieved by providing video monitors of varying sizes, shapes, or phosphor characteristics. Some examples of how this modularity could be employed to simulate various conventional display systems follow.

Figure 7 shows a system configured to support a typical NTDS sensor console. A single high-resolution monitor is employed, receiving inputs from two sources: an OTF channel which produces the cursor, tactical symbols, geographical symbols, lead vectors, alphanumeric characters, and other dynamic graphic elements; and a radar simulator, producing the raw sensor information. The heavy dashed line indicates

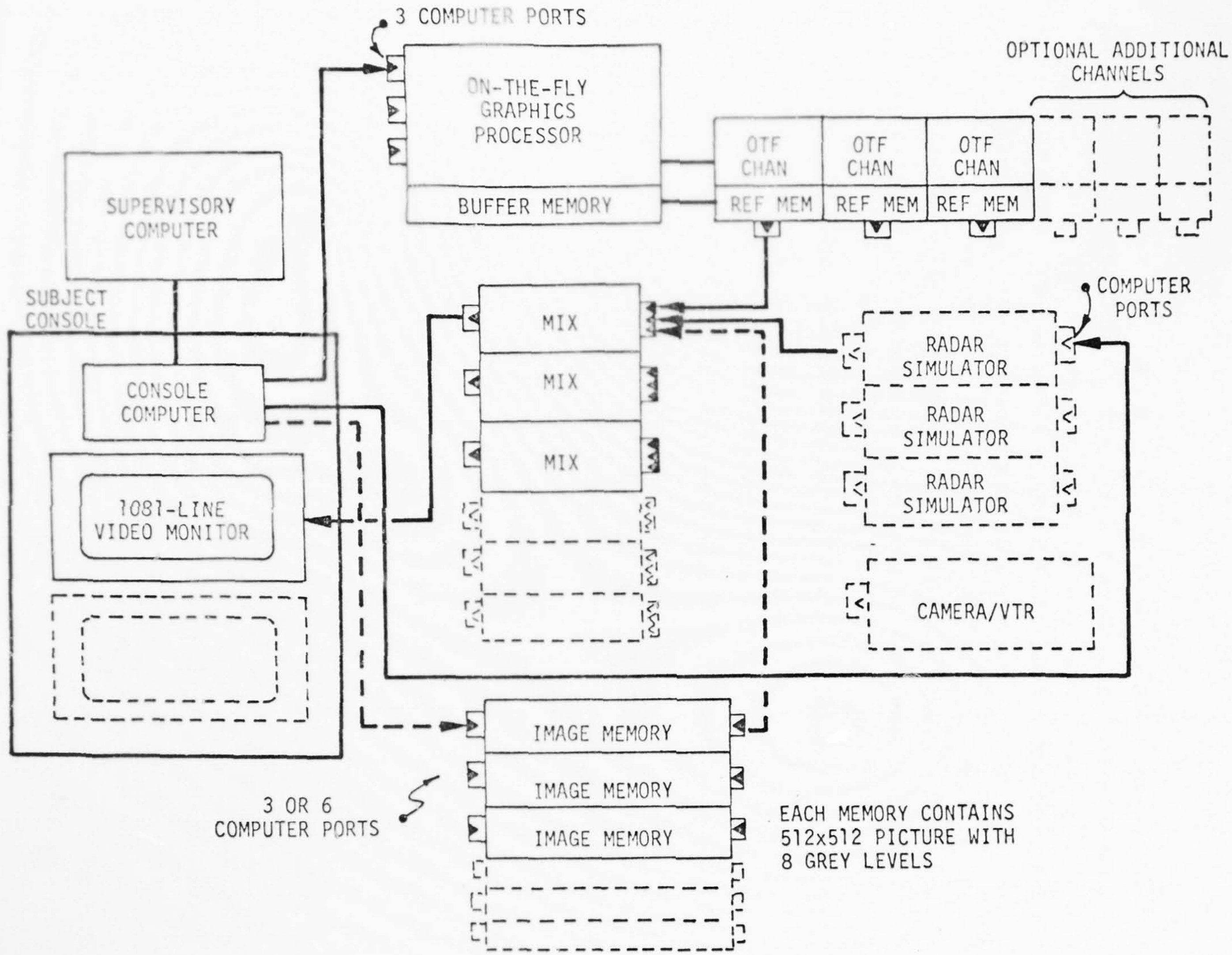


Figure 7. TDS simulation.

that an image memory might be employable instead of a radar simulating device. We suspect that, by substituting a long-persistence video monitor, an image memory could be employed to produce a convincing simulated radar PPI display. However, while we have in the course of this study mocked up a scan-converted radar display and satisfied ourselves of its suitability, we had no means for experimentation with an image memory. Consequently, we are uncertain of the validity of this idea and must therefore put forward a radar-simulation device as a tentative necessity. We consider it very desirable to test the feasibility of the image memory approach as early as possible during the MSRF development effort.

The reader will note that in either case only 1/3 of the recommended modules are employed in this configuration; therefore, three such displays may be independently created.

In Figure 8 we see these modules organized as would be necessary to simulate passive sonar displays. Once again, the dynamic data, such as cursors and alphanumerics, are produced by an OTF channel while the dense, relatively static processed sonar data is presented in grey scale by an image memory. This assemblage of 1/3 of the recommended equipment is required for each *independent* sonar display; thus, if each console were equipped with a single monitor, three consoles could be supported.

Unfortunately, most current sonar consoles are equipped with *two* independent displays. While this can be supported by the recommended system as illustrated in Figure 9, it is easy to see that each such console requires the dedicated services of two of each of the basic modules or 2/3 of the resources of the recommended system. Thus, while one such console could be simulated, the remaining modules could only handle the requirements of a single-display sonar, or perhaps an NTDS, console. To configure the system for three such

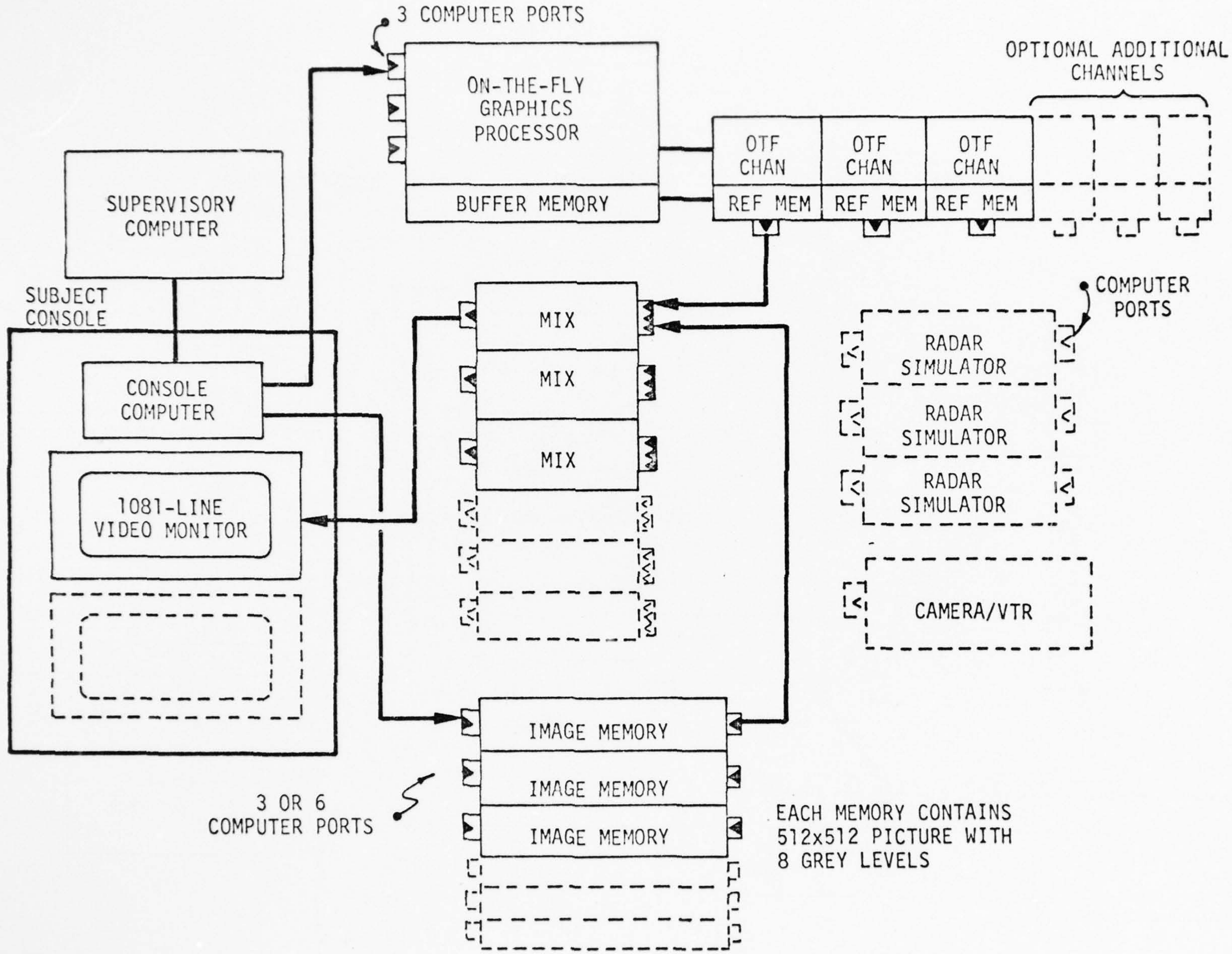


Figure 8. Single display sonar simulation.

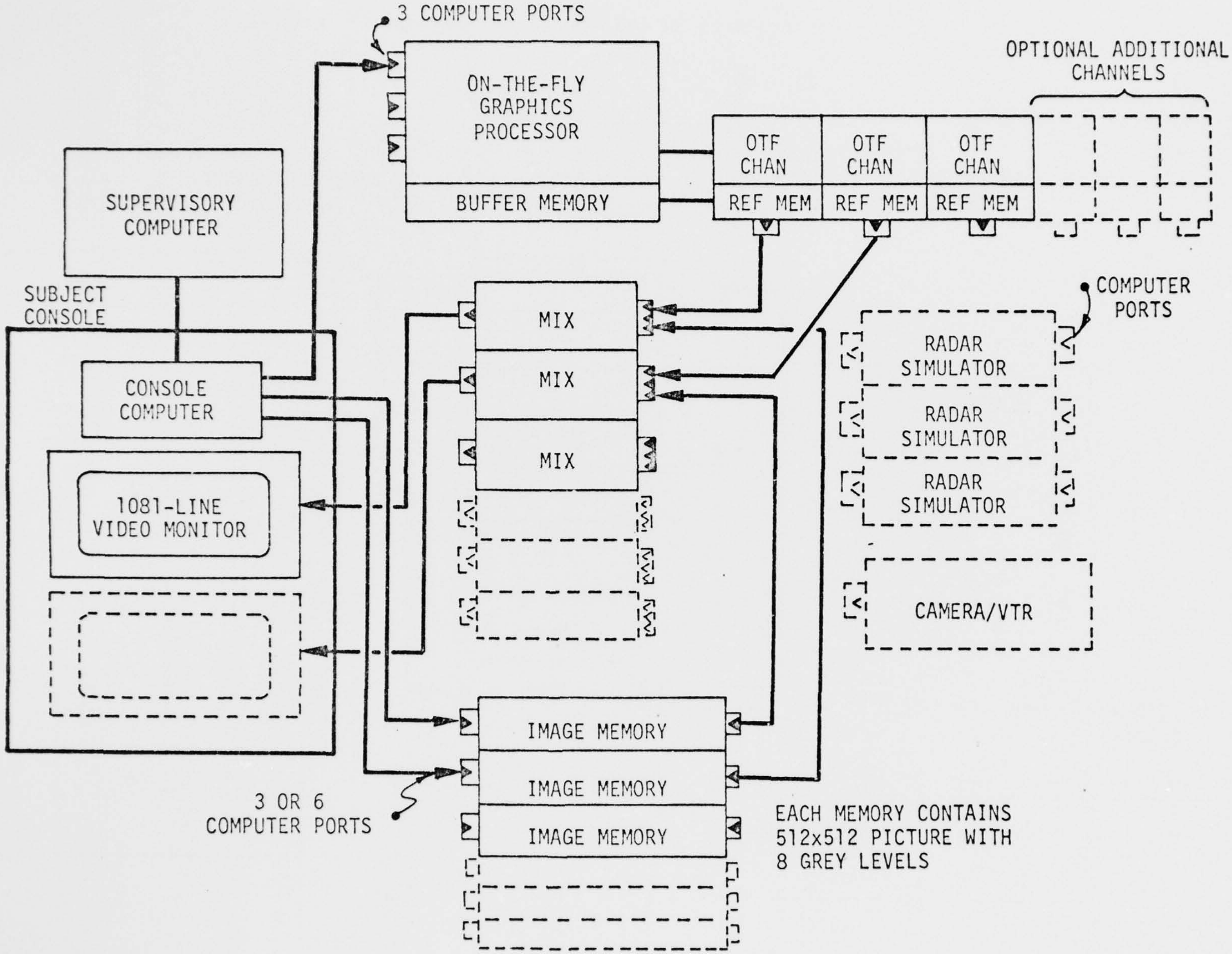


Figure 9. Dual display sonar simulation.

consoles, it would be necessary to include the optional modules, increasing the price of the system by roughly 50%.

Finally, in Figure 10, we see a configuration where the display is composed of a prerecorded moving picture upon which dynamic graphic elements may be superimposed. While this and other potential applications involving imagery were not mentioned in the original requirements for the MSRF, the capability is a no-cost fallout of the recommended techology and, as such, should not be ignored. Perhaps a more practical application is suggested by the observation that photographic slides and other visual aids could be displayed by employing available devices.

As of this writing, we believe that the best choice of display systems for the MSRF would be a configuration of Hazeltine modules such as we have just described, although it is entirely conceivable that within even the next 6 months other vendors may announce similar products which could be superior or could cost less. This possibility should be explored before any procurement is initiated; however, we would caution NPRDC that the evaluation of such a system as this is neither quick nor easy. Even if Hazeltine were chosen as the vendor, the exact specifications mentioned earlier such as memory sizes, image memory resolutions, options, and other details should not be simply quoted verbatim. In the context of this study, our discussions with Hazeltine were limited by the uncertainty of a procurement; certainly the expenditure of something like 2 weeks in additional negotiations could bear substantial fruit in improvements of the overall configuration or reduction of price. Hazeltine has furnished informal quotations indicating that the recommended system could be delivered for a total of $250,000 and that delivery would be approximately 6 months ARO if an order were placed in the Fall of 1977.

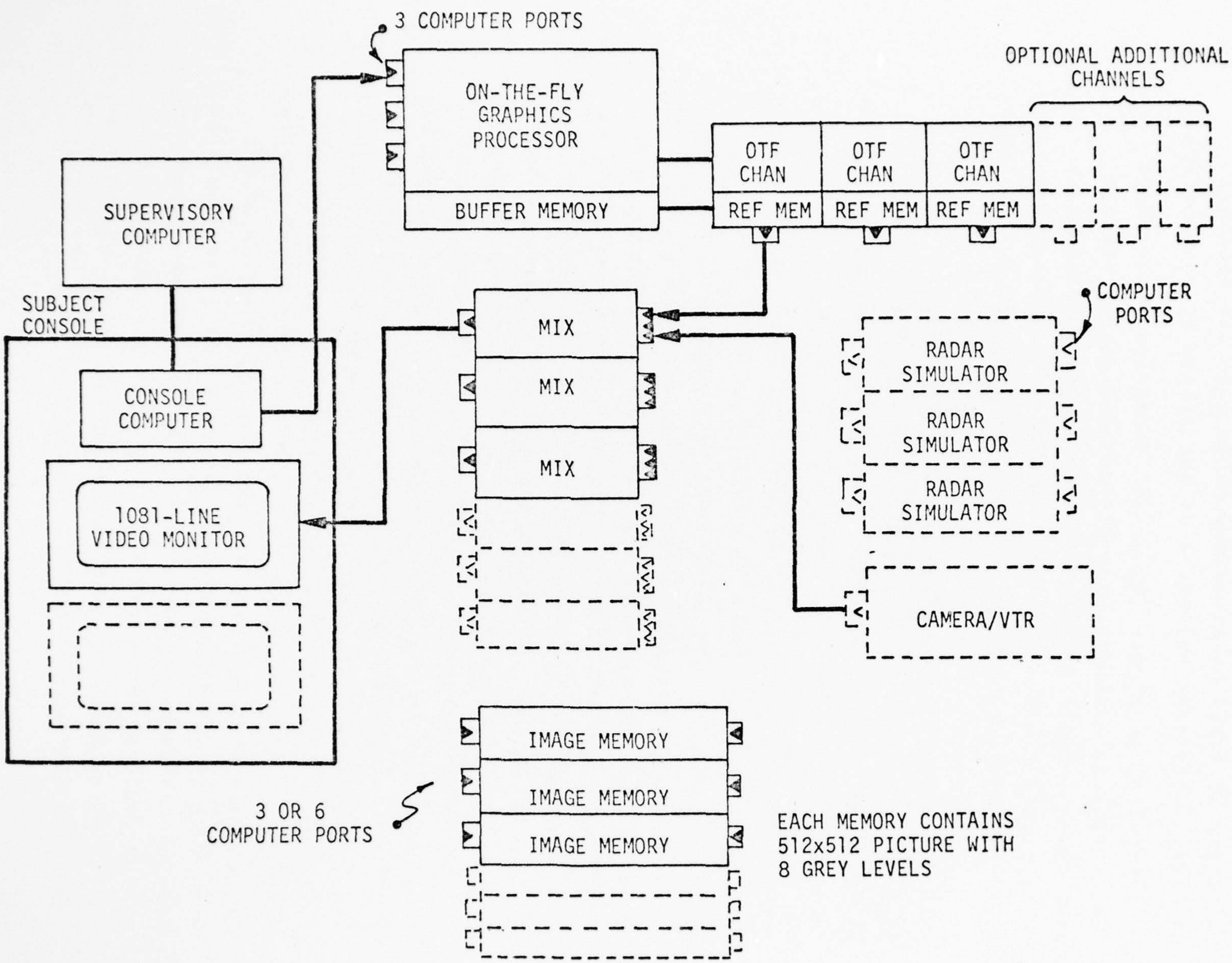


Figure 10. Imagery system simulation.

Console Controls

Very few operator input devices are required for the MSRF console in the primary NTDS and sonar configurations. Essentially the same functions available on the UYQ-21 console should be provided, with the possible addition of keyboards. All functional requirements can be met by the use of push buttons and a cursor control device. Using variable function buttons with variable legends and a numerical keypad would obviate the need for most other types of buttons or switches. Either a trackball or joystick can serve as the cursor control. The recommended operator input devices are discussed below.

VARIABLE FUNCTION AND LEGEND BUTTON MATRIX

A button connected to a computer can have different functional meanings depending upon the state of the system. If a number of options are available to the operator, as is typical in NTDS and sonar systems, the operator must be aware of the current functional meaning of a particular button. In the UYQ-21 console, the legend change is provided by an opto-mechanical button matrix which projects images from film chips on the button surfaces. Twelve to 24 legend options are available for each button in this device. Unfortunately for research purposes, it is difficult and time consuming to change the film chips. A better method would allow an indefinite number of legends to be presented on each button. Such a capability exists using a plasma touch panel. This device consists of an 8 x 3 inch plasma panel surrounded by a thin frame which contains light-emitting diodes and light detectors, so that an interruption of the light beam by a finger can be sensed. The photodiode array is a matrix of 3 x 16 elements. Since a typical push button is a square 1 inch on a side, 24 virtual buttons can be implemented on a plasma touch panel. Sixteen rather than eight diodes are used along one dimension of the panel to allow change in the

relative positioning of the virtual buttons. Within a given button area any desired legend can be created under computer control. Current sonar and NTDS legends never exceed three lines of six characters per button. Since the plasma panel has 60 elements per inch, eighteen 9 x 16 point characters can easily be accommodated within the space of each virtual button and the outlines of the buttons themselves may be drawn by the computer. The plasma panel and diode matrix are under computer control and therefore legends can be created or changed and button presses sensed by use of appropriate software. The touch panel is a tremendously powerful and versatile device for the MSRF console since the type of lettering, symbology, and shape of buttons can be easily changed. Because the panel is flat and has a total depth of less than 2 inches, it can be placed almost anywhere on the console. It is recommended that two such devices be included for each console. A single 24-button matrix would probably be sufficient for most sonar and NTDS applications, but use of two of the devices would allow additional capabilities by providing more options to the subject/operator at a given time. Besides use strictly as a button matrix, one of the plasma touch panels can be used as an auxiliary information display device.

The plasma touch panels will interface to the LSI-11 console computer and both panels can be accommodated by a single interface card in the LSI-11. These devices are available from Information Technology, Limited (ITL), 706 Jackson Street, Monticello, Illinois 61856. Two 3 x 8 inch touch panels and the interface card will cost approximately $10,000. This price is comparable to the cost of the opto-mechanical button matrix incorporated in the UYQ-21, but is much more versatile.

VARIABLE FUNCTION BUTTONS

In addition to the plasma touch panel button matrices, two columns of momentary contact buttons will be provided for

use along the side of each CRT display. Buttons of this type are available from a number of sources. For example, Micro Switch and Switchcraft supply buttons suitable for this purpose at \$10-\$15 each, with provision for internal illumination to focus the operator's attention on a single button or group of buttons.

FIXED FUNCTION BUTTONS

Forty-two additional buttons similar to those used alongside the CRT screen will be available for inclusion in the console. In the UYQ-21 console, these buttons are configured in a 6 x 7 matrix located on the lefthand side of the bullnose. These buttons will be compatible for use in a matrix or may be placed individually at different locations on the console. These buttons are also available from the above sources at approximately the same price.

CURSOR CONTROL

All Navy systems, including NTDS and sonar, which show symbolic information on a CRT screen require some means for identifying, originating, and removing the symbols from the screen. Usually a moving spot of light or a symbol, the cursor, is controlled through a joystick or trackball. Both of these devices operate in a similar fashion, providing for X and Y translation of the cursor which tracks corresponding movements of the operator's hand. Although there are some basic differences in functional capability, the choice of one over the other is usually dependent on some minor human factors consideration. It is our recommendation that both a trackball and joystick which are interchangeable be used for the cursor control in the MSRF console. In addition, an "ENTER" button adjacent to the trackball or located on top of the joystick will be necessary to signal the computer that the cursor is properly located for the desired action. Trackballs and joysticks are often available from the same company. For example, Measurement Systems, Inc., of Norwalk, Connecticut,

offers suitable joysticks for approximately $1,170 and a trackball with the appropriate output signals for approximately $1,600.

ADDITIONAL CONTROLS

In addition to these controls, the CRT tuning controls will be on the console. Their number and location will depend upon the display system used. A communications panel is usually included on display consoles also. The communications controls will be discussed in a later section. The controls listed above will provide all the necessary functions for simulation of NTDS and sonar consoles. Additional devices that may be necessary, for example, for an integrated bridge control mock-up, would involve additional small devices such as thumbwheel switches and rotary switches. These same functions, however, can probably be implemented using plasma touch panel displays. For other configurations such as a propulsion system, discrete panel meters and minor devices would be required. These devices are readily available at low cost from a number of sources and can be easily obtained at the time need arises. Interfacing of all of these devices will be relatively simple because of the use of a common interface which can provide and accept both digital and analog signals. Full ASCII keyboard modules, available from a number of suppliers, should be provided to serve the needs of experiments requiring alphanumeric input and to enable use of a subject console as a programming and checkout terminal.

Requirements for secondary alphanumeric displays can be met in several ways. As was mentioned earlier, one of the two plasma panels could be devoted to this purpose. Self-scan panels offer displays of limited size at low cost. Modular, alphanumeric raster-scan TV display generators are available from several sources, such as Ann Arbor Terminals,

Ann Arbor, Michigan. Whether any of these devices should be procured within the first 3 years should depend on the requirements of the initial experiments, since they are inexpensive and readily available at any time and since their presence or absence has no particular effect on the overall system design.

The common interface will permit acquisition and use of less generally applicable devices to meet the requirements of particular experiments.

Console Framework

It must be possible to reconfigure the MSRF console to simulate the appearance of a variety of Navy system display consoles. Within a given framework it is simple enough to relocate components, but the basic framework must also be capable of reconfiguration. Standard equipment racks would not be suitable for use as the MSRF console framework for this latter reason, and because their standard width is not compatible with the sizes of typical Navy consoles. Custom-sized racks could be built but the cost would be high and they would have very little flexibility for change. Fortunately, a highly flexible and low-cost means for constructing the MSRF console framework is available. The Widney-Dorlec 20/30 Constructional System is a line of compatible frame parts which allows the construction of consoles of almost any desired size or shape. The primary components of the system are various sized metal extrusions and cast corners that bolt together. Smaller pieces of metal bar can be bolted to any point on the frame for supporting the console instruments. The sides of the console can be disconnected to allow reconfiguring the individual consoles into one large team console.

This system has been approved by the British Ministry of Defense for use on Royal Navy ships. The component pieces

are extremely sturdy and in the MSRF simple materials such as sheet aluminum or masonite can be used for the paneling on the front, back, and sides. Using this system it will be extremely easy and fast to configure consoles for each experiment.

The displays, buttons, trackball, etc., would be placed in self-contained modules that can be located anywhere within the console framework. The console front can be given a finished appearance by filling in any holes in the console face with Foam Core.

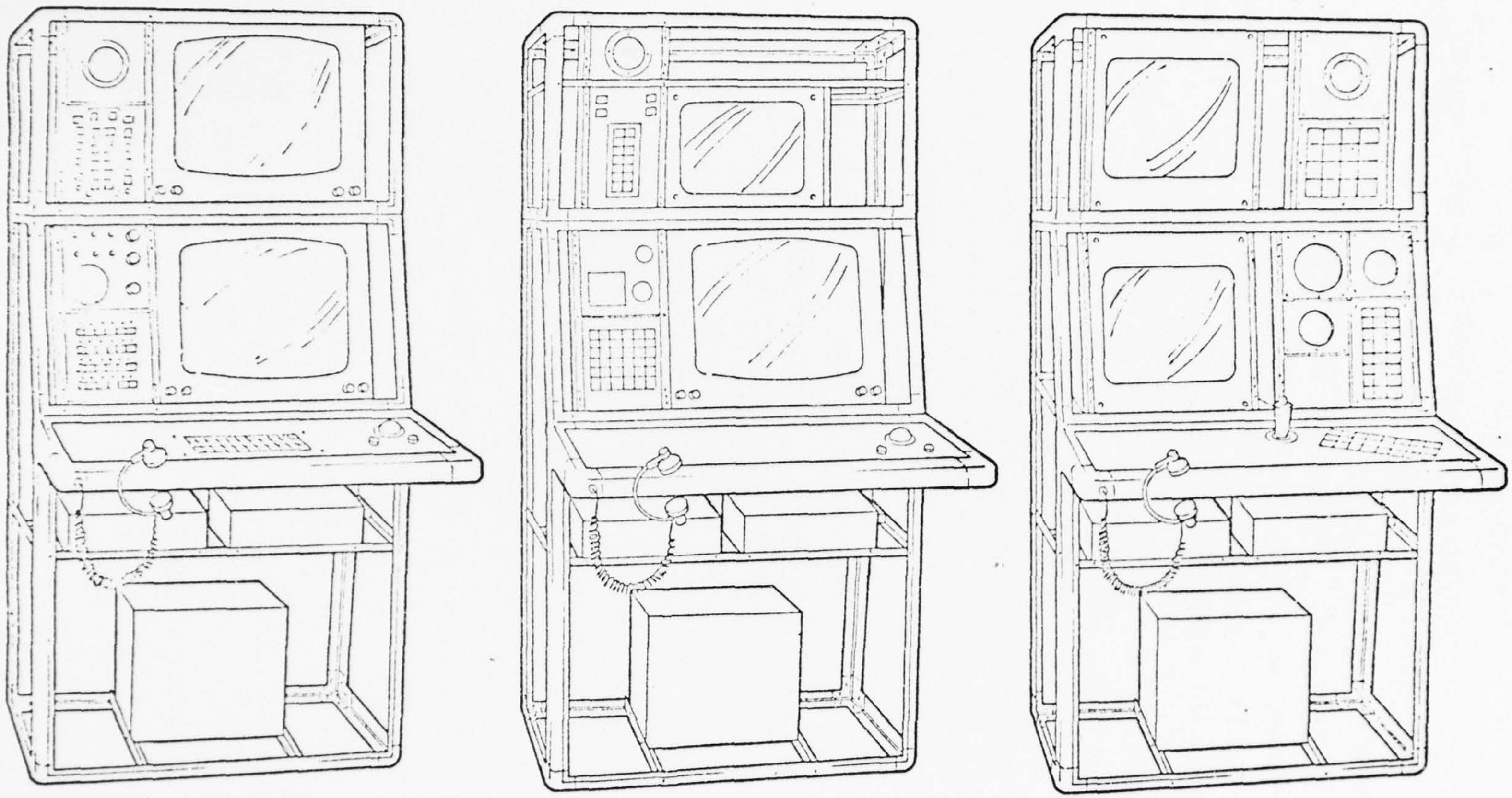
Figure 11 shows how modules may be flexibly mounted within a given framework shape to produce consoles similar to the UYQ-21 in sonar and NTDS configurations and to the BQQ-5 sonar console. The illustration shows use of display CRTs of different sizes commensurate with those used on the actual consoles; one of the advantages of the video approach to graphics generation is the ability to substitute relatively inexpensive (about $2,000) video monitors having differing characteristics without otherwise altering the graphics subsystem.

While all three frameworks illustrated have the same general shape, one need only remove the modules and unbolt the frameworks at their corners to achieve different shapes.

The components for the framework are available through the Dorlec Corporation, Cherry Hill, New Jersey. The components for each console will cost approximately $500.

Console Computer and Interfacing

Thus far we have specified consoles which consist of a flexible physical framework in which a selection of functional modules may be mounted to represent the operator work stations of a variety of systems. Each of these modules was selected on the basis of its ability to adapt to multiple functions.



UYQ-21 SONAR UYQ-21 NTDS BQQ-5 SONAR

Figure 11. Representation of console modules configured for three different applications.

This adaptability was in most cases achieved by two means. First, the modules selected contain a minimal amount of information processing or feedback capability so that each has little intrinsic behavior beyond that of a pure transducer. Second, this missing behavior is implemented in a digital computer system which can readily be reprogrammed to provide different behaviors (software-intensiveness).

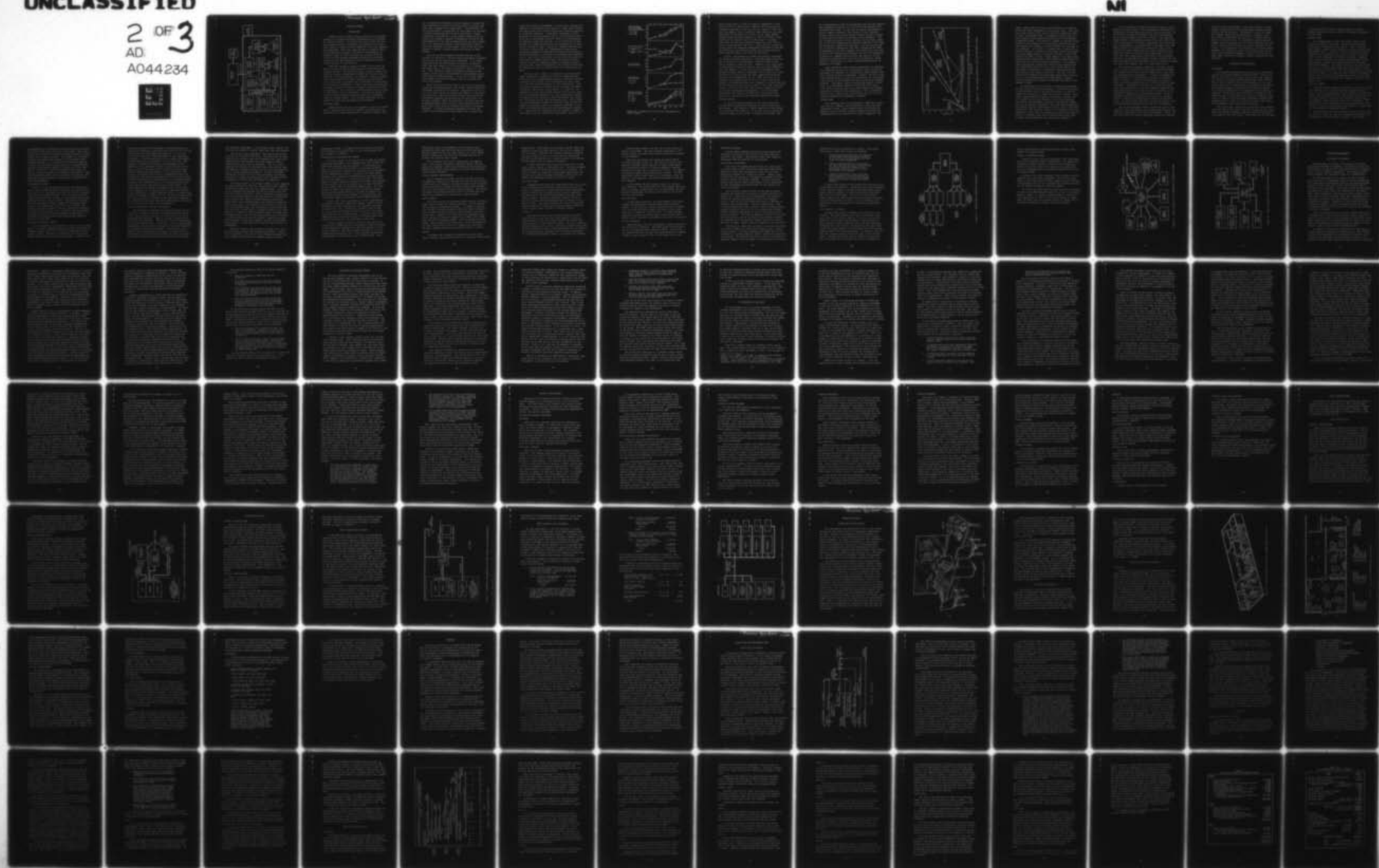
One thing which characterizes such a software-intensive system is the intimacy of contact between the computer and the peripheral devices. More lines of communication are required, and more information must pass over those lines in order that the computer may perform work which would otherwise be done by the peripheral devices. The volume of this intercommunication, and the work it implies within the computer system, presents a very real design problem which we propose should be solved by extending a portion of the computer system into each console. Of the various ways in which this could be done, the most attractive from the standpoints of power, flexibility, symmetry, and availability involves the placement of a microcomputer within each console to interface with display and control elements, implement their behavior locally and, thus, vastly reduce both the number of devices attached to the main computer and the processing burdens imposed on it. Figure 12 shows how the modules in a single console would fall into the hierarchy of such a system.

The console computers themselves, as well as the methods used for interconnecting them with console modules, will be discussed in the next section; for, although they are physically located inside the consoles, they must be regarded conceptually as part of the computer system.

AD-A044 234 HUMAN FACTORS RESEARCH INC GOLETA CALIF F/G 5/8
FUNCTIONAL REQUIREMENTS AND OTHER DESIGN FEATURES OF A MANNED S--ETC(U)
AUG 77 G V BAILEY, R T HENNESSY, C D WYLIE N00123-76-C-1245

UNCLASSIFIED NL

2 OF 3
AD
A044234



SIFIED

2 OF 3

AD
A044234

1.0 2.8 2.5 3.2 2.2 3.6 4.0 2.0 1.1 1.8 1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

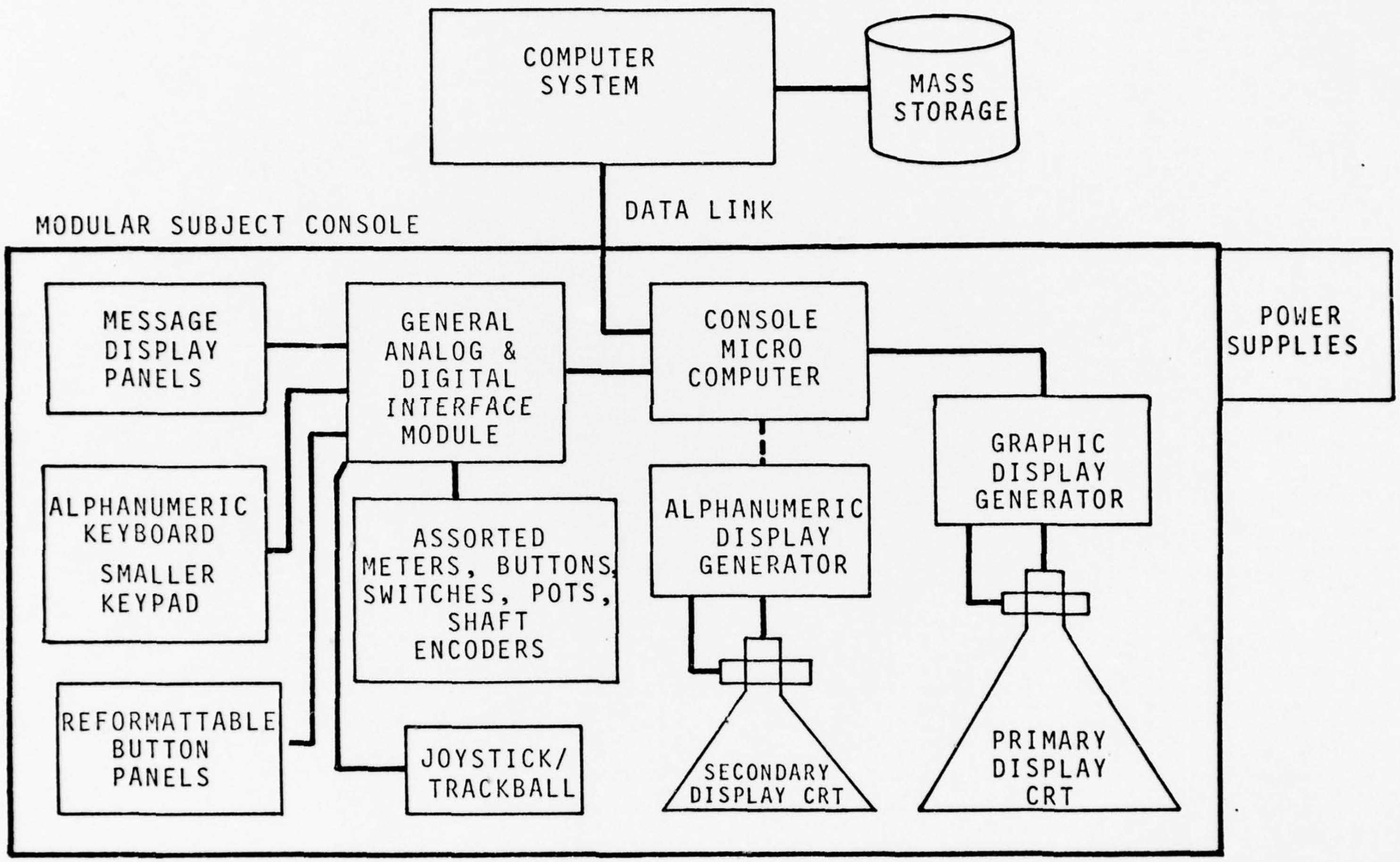


Figure 12. Relationship of console modules.

Preceding Page Blank - Not Filmed

COMPUTER SYSTEM

Background

Perhaps the key element lending flexibility to the MSRF design is its software-intensive implementation of virtually all console features. The behavior of each control and display element used for interfacing the subjects to the simulated system is almost entirely dependent on how the computers are programmed. While this approach leads to greater flexibility than does any other with which we are familiar, it naturally tends to shift the burden for success of the facility onto the computer equipment selected and, more specifically, onto the shoulders of those responsible for producing software for the facility.

While the conceptual mission of MSRF is fairly well defined, the full technical ramifications of that mission are somewhat elusive. One could say that in order to support the kinds of research contemplated for MSRF, the facility must be "powerful, easily programmed, and flexible." However, quantitative specifications of these attributes are unavailable, since the specific experiments to be conducted in the facility are by necessity undefined. In fact, were such specifications available, a precise statement of requisite capability could be formulated. In the case of a general-purpose facility such as MSRF though, this is impossible. Instead, the design effort has centered around *maximizing* power and flexibility within an assumed budget. The following discusses each dimension in which this optimization was performed.

COMPUTING POWER

Computing power can be a very difficult thing to quantify. Given a specific application to analyze, and given that the application has been programmed for a specific computer, then

it is possible to precisely state the computer's "power" *for that application* in terms of the percentage of available machine time consumed by the application. Alternatively, if the quantification is being done for the purpose of selecting a general-purpose machine for batch processing, a fairly satisfactory statement of a given system's power may be arrived at by running a lengthy, heterogeneous stream of jobs representative of the work contemplated for the batch system. Yet, even though these methods yield quantitative results which are acceptable to many people, such results are still subject to several variables beyond the control of these methods.

Each of the above generally accepted methods of measurement is sensitive to the particular programming techniques used in each of the applications considered. It is widely recognized that the time required to execute a program is often much more sensitive to the efficiency of its design and coding than to the type of hardware on which it is to be run. Thus, empirical methods of computer performance measurement often result in measurements of programming quality rather than of computing power. In the case of MSRF, empirical methods are excluded entirely since no relevant test cases are available.

While much work has been done in the area of theoretical estimation of computing power, the results obtained from such techniques leave much to be desired. The general idea is to list the times required to execute each member of a machine's instruction set, and then to multiply each such time by the anticipated frequency of execution of the instruction in a "typical" program. While the individual execution times may be determined with great accuracy, gross misconceptions of power often result from the inherent inaccuracy of frequency estimates. If machine language programming is being considered, these frequencies seldom reflect the savings which

result from "clever" programming. If high-level language programming is anticipated, the frequencies almost never reflect with any accuracy the overhead instruction sequences generated by optimizing compilers. Furthermore, regardless of what mechanism is used to generate code, no set of frequency estimates will be meaningful unless it takes into consideration the number of registers available, or the difficulty of performing such common operations as array indexing or the testing and setting of flags. Generally, computers whose characteristics in these areas are "inhospitable" require that considerable numbers of overhead instructions be executed or that great deftness be exercised in overcoming them programatically. What all of this means is that anyone who claims to have a pair of numbers which reflect the overall relative "powers" of any two computer systems of differing architectures has most likely oversimplified the problem so much that his numbers may only be trusted within the conditions of his tests.

The only case in which it is feasible to generate meaningful quantitative estimates of relative computing power occurs when one is comparing processors of identical internal architecture and roughly linear differences in their execution times for each instruction in their repertoires. Such a comparison between various models of the Digital Equipment Corporation (DEC) PDP-11 series appears in Figure 13. An examination of the curves should acquaint one with the difficulty of comparing computers; for, even though such a family of computers should present an ideal environment for comparisons, the curves do not track one another well. While there is little difference between the 11/40 and 11/45 in terms of simpler instructions, there is a dramatic difference in execution of integer multiplication and floating-point arithmetic. Looking at this comparison for the purposes of MSRF, it would appear that the unquestioned performance leader in all

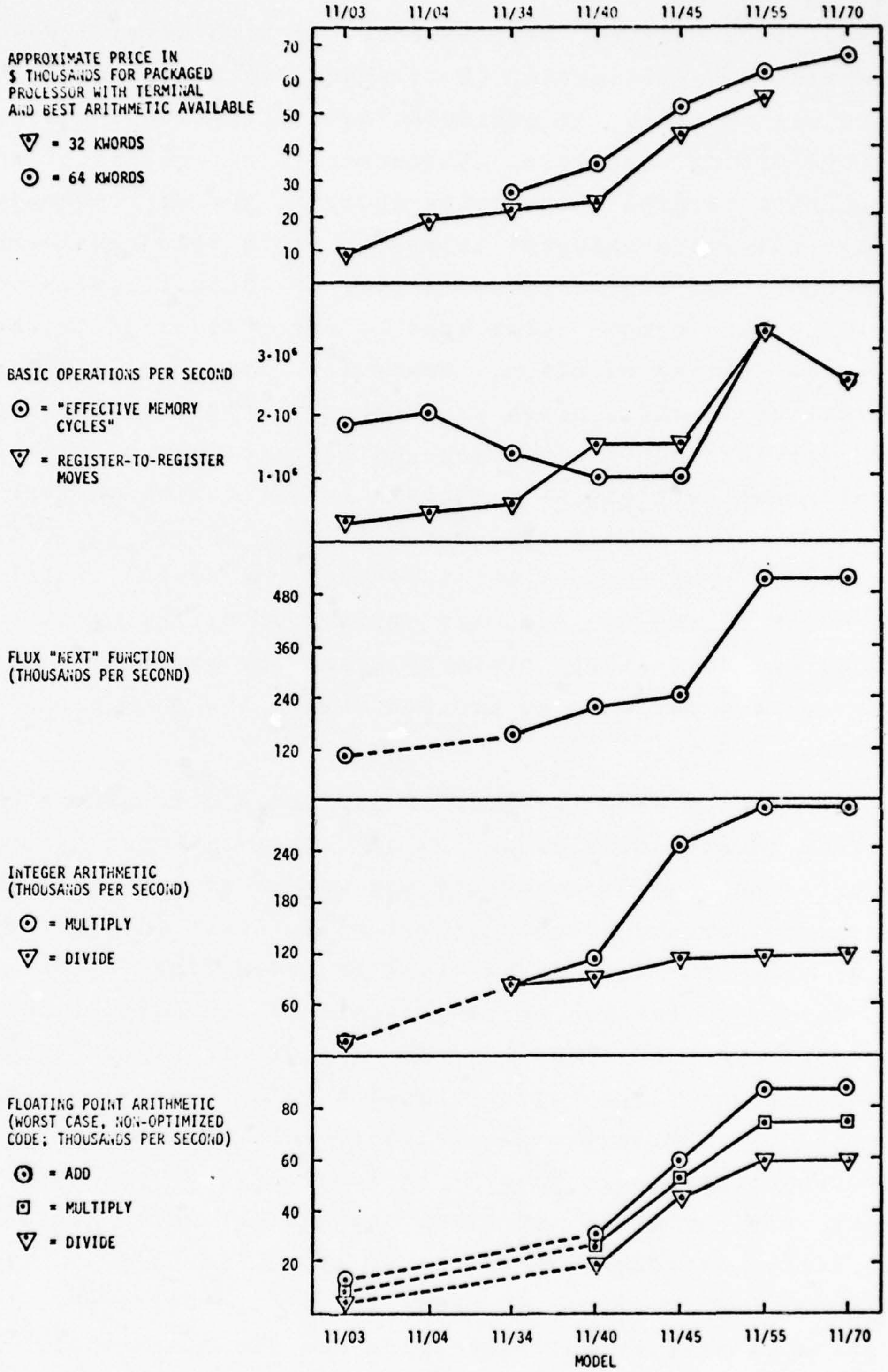


Figure 13. Comparison of price and performance for PDP-11 models.

areas is the 11/55, in that its speed is comparable to that of the 11/70 at a lower price. However, this conclusion is somewhat misleading in that the 11/55 is simply an 11/45 with very fast memory; only a relatively small amount of this fast memory may be installed in an 11/55, so for larger memory configurations its speed degrades to that of an 11/45. Thus, looking at the slopes of all of these curves, it becomes clear that the most cost-effective processor in the line is also the most powerful, the 11/70. Given that all we know for certain about the computing requirements for MSRF is that they may potentially be arbitrarily large, then the processor of choice from the PDP-11 line would be the 11/70.

In fact, circumstances at NPRDC considerably simplified the choice of central processors. The ATTS computer facility was at the point of "outgrowing" its DEC PDP-11/45 system and was hoping to order an 11/70. Normally, this would result in surplusing of the 11/45 with all the costs and inefficiencies that implies. It was clear that the government would maximally benefit from its investment in the 11/45 if it could be placed in service within the same command which had previously held accountability for it. While the 11/45 is undeniably less powerful than the 11/70, it must still be regarded as a powerful machine; and, although its capabilities may be more easily exceeded than may those of the 11/70, experience leads us to believe that the 11/45 would be adequate for all but the most demanding investigations. We have therefore recommended that MSRF use the PDP-11/45 being released by ATTS as the overall most cost-effective approach for NPRDC.

STORAGE CAPACITY

It is clear from the results of the survey discussed earlier that a considerable quantity of online storage must be provided. Disk storage is needed regardless of application as a repository for software. It is also the best medium for interim storage of data collected during an experiment; HFR

has experimented with the use of magnetic tape for this purpose, with disappointing results in terms of both reliability and attainable data rates. In the case of lengthy experiments involving complex scenarios, disk storage is also indicated. Finally, some of the survey responses showed interest in maintaining large data bases on the order of two megabytes. Figure 14 shows the results of an analysis of price versus capacity of the disk storage devices available from DEC at the time the supervisory computer was specified. HFR has noticed that even in simpler environments than that contemplated for MSRF, 20 megabytes can be quite cramped; this observation, when combined with the foregoing requirements and with the information in Figure 14, suggests quite directly the choice of the 88 megabyte disk-pack drives as the most cost-effective, being priced at (curiously) less than a system offering merely 20 megabytes of storage.

Large simulations often require maintenance of large data bases defining the state of the simulated system, which must be updated or referenced at a sufficiently high frequency that their storage on disk is infeasible. Further, the size of the algorithms necessary to drive such simulations is often large. A core memory of 196 kilobytes has been specified as probably adequate. Once again, without assuming some specific simulation, it is regretably impossible to justify these exact sizes beyond asserting their cost-effectiveness and probable adequacy on the basis of our experience. This is especially true with core memory since the efficiency of its use is intimately dependent on programming techniques.

SURFACE AREA

The computing system to be embedded in MSRF may be characterized as a complex, real-time process control and data acquisition system. These characteristics are further exaggerated by the software-intensive approach advocated for implementation of the functions of console components, and

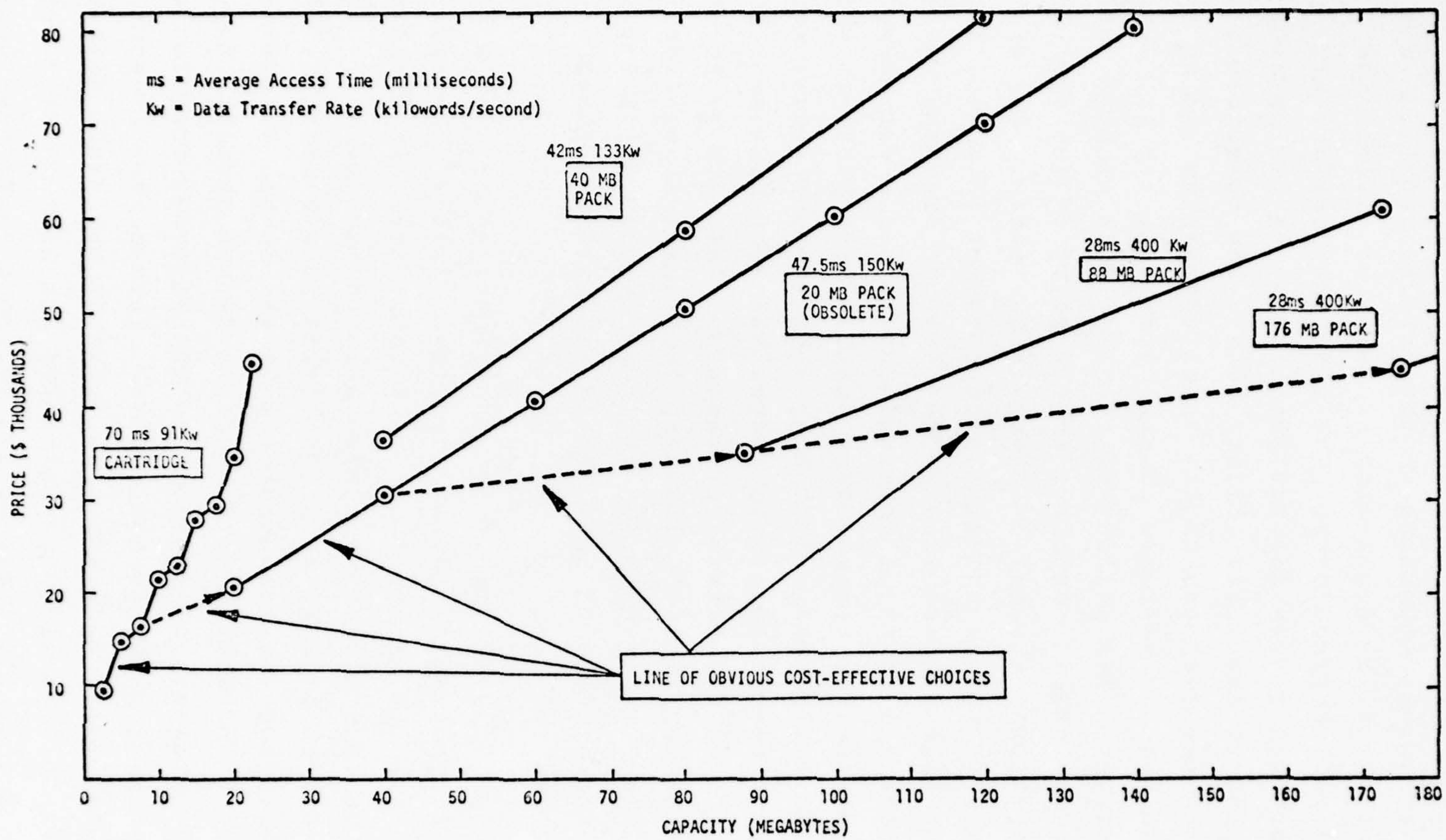


Figure 14. Mass storage capacity vs cost for DEC PDP-11 units.

imply a great deal of intimate contact between the computer system and things happening in the external world. This intimate contact may be dealt with in terms of a dimension which we shall call *surface area*. The effective surface area of a given computer system depends principally on its peripheral interfacing scheme. Its ability to respond to external events depends on the richness of its interrupt system and on the time required for software to switch context for servicing of the interrupt. Its ability to transfer data at high rates to multiple devices is sensitive to the power of the direct memory access (DMA) channel or channels available. One of the greatest problems in using a large computer in an application demanding high surface area is the extreme overhead which can result from all of these external interfaces. For example, large time-sharing systems have surface area problems which are sufficiently severe that most such systems employ communications concentrators to reduce the responsiveness burdens placed on the central processors. The idea is to *distribute* the peripheral processing load among several devices, and it is our recommendation that MSRF employ a similar distributed-processing approach to maximize the power of its computer system.

The principal foci of external interfacing within MSRF are the subject work station consoles. If the PDP-11/45 were to deal directly with all external interfaces, it would assumedly be connected by a large cable to each console. Every time a subject pressed a button or adjusted a control, the 11/45 would need to stop whatever it was doing, determine and execute the appropriate reaction to the operator input, make any necessary alterations to the state of the simulation, and possibly log the transaction or extract from it some data element for later analysis. While it is certainly within the capability of the 11/45 to multiplex its time in this fashion, there is a definite limit to its responsiveness when there are many such occurrences to be dealt with frequently.

A very reasonable alternative would be to place a microcomputer in each console. The microcomputers would have responsibility for implementation of all *console functions*, in the sense of closing all feedback loops with the operators which logically happen within the consoles. This approach would relieve a considerable portion of the work load of the 11/45, permitting it to assume a *supervisory* role in the conduct of experiments. A typical task of the supervisory computer would be to simulate the *system* under study, by receiving only those inputs from the console computers which affect the system as a whole and disseminating to the consoles only those transactions which affect their operation. Naturally, all information transfer between consoles, and all data collected from the experiment, would be channeled through the supervisory computer, but at a much reduced rate.

The transaction handling capability, and hence surface area, of such a distributed system should extend the effective capabilities of the 11/45 far beyond those of an 11/70 in a non-distributed environment. To select an appropriate microprocessor, it is necessary to consider its probable tasks. Clearly, it would be self-defeating to specify a micro with serious deficiencies in its own interrupt or DMA facilities. Furthermore, the micro to be used must offer relatively high power in areas of arithmetic capability and memory addressing, and it must be easy to interface to the supervisory computer and to peripheral devices. These requirements effectively exclude all of the 8-bit microprocessors, since they are deficient in several of the above areas. The only viable contenders at this time appear to be the Data General MicroNova and the DEC LSI-11 computers.

Each of these machines offers instruction times on the order of a few microseconds, with integer multiplication and division on the order of 40 to 60 microseconds. However, we feel that there are several compelling reasons for choosing

the LSI-11 over the MicroNova. The latter employs a fairly archaic instruction set with limited memory addressing capabilities and few registers. Furthermore, it does not offer any hardware support for floating point arithmetic, while this feature is available for the LSI-11. Perhaps more important is the point that in a distributed processing environment such as we propose, it is vitally important that software be *easily movable* between the various computers in the system. If the MicroNova were selected as the console microcomputer, this movability would not be possible for any coding written in machine language. The LSI-11, on the other hand, offers practically complete software compatability with the 11/45.

Recommended Configuration

PROCESSORS

The foregoing considerations have led us to the specification of an all-DEC computer system for MSRF. There is in fact no real choice in the case of the supervisory computer; and, while it is obvious that microprocessors other than the LSI-11 could be used for the console computers, it is also obvious that such a substitution would complicate the facility's software support considerably and ultimately detract from system performance by discouraging machine language programming. We feel that this combination of computers will probably prove to be adequate for the needs of MSRF for many years, so long as those who design experiments remember that computing power is after all a finite resource, and consult with programmers about their plans accordingly.

Thus far we have confined our discussion of the computer system to processors, memory, and mass storage devices, since these are the most important dimensions in which sufficient raw capability must be assured. While MSRF does not need many

peripheral devices which are commonly found in general-purpose computing facilities, several additional ingredients are necessary.

MAGNETIC TAPE

We recommend that at least one 9-track, 800 cpi tape drive be attached to the 11/45. There are two principal motivations for specifying this device. First, since the MSRF computer system is not expected to reduce or analyze the experimental data it collects, some fairly universal storage medium must be available so that these data may be communicated to other computers for further processing. Additionally, it may be desirable to accept input data generated by other computer systems (as suggested by some of the questionnaire responses). Nine-track, 800 cpi is the most widely used of the available tape formats and would offer the maximum capability in this area. The second motivation is strictly internal to MSRF; it is essential that a means be provided for backing up all or part of the programs and data on the large disk on some easily stored medium. Magnetic tape serves this purpose well.

HARD COPY OUTPUT

The MSRF would be highly inconvenient to use if it lacked *some* capability for producing hard copy output directly. While interactive programming over CRT terminals is generally viable, most programmers become quite inefficient if they cannot easily obtain listings of their programs, memory dumps, or diagnostic output in a form which can be marked up. The need for hard copy documentation is easily perceived as well. Finally, it may be very desirable that summary or logging data for an experiment be produced directly.

There are many possible hard copy devices to choose from, but most are either too slow to do programmers much good, or too expensive for these needs to justify. Our specific

recommendation is the Printronix 300 line-per-minute printer, which uses normal paper and costs about $5,000. This seems to be about the most cost-effective medium performance printer around, and is highly reliable. Furthermore, it is capable of lowercase printing and graphics, and can produce quite readable documents. A less expensive alternative might be a device like the Diablo HiTyper which is a changeable-font typewriter-like machine capable of speeds up to four times that of a Selectric. These cost about $2,500 without keyboards, and are available with standard EIA interfaces. They are considerably better than the printers for producing documentation, but their speed is marginal for satisfying the needs of programmers.

EXPERIMENTER CONSOLE

While it is not anticipated that the experimenter will need a console enabling him to orchestrate the experiment in real time, it is inescapably necessary that he be in communication with the computer system. It is our opinion that a single CRT terminal can satisfy any such communication needs, provided that the terminal chosen is sufficiently flexible. We would recommend a terminal capable of displaying around 24 lines of 80 characters, with full cursor control, and having a full ASCII keyboard with some reasonable complement of special function keys. It is essential that the terminal be capable of true full-duplex operation at 9600 baud. A terminal which would satisfy these requirements admirably would be the DEC VT52.

SOFTWARE DEVELOPMENT CONSOLE

Strictly speaking, the experimenter console (or, for that matter, a suitably configured subject console) constitutes an adequate terminal device for interactive programming of the computer system. Initially, we anticipate that the experimenter console would be used for this purpose. However, such

usage would prevent the development of software during the conduct of experiments, and would tend either to force strange working hours for programming personnel or to extend the set-up time required to configure MSRF for each experiment. Ultimately, it should become possible to conduct development and experimentation concurrently. At the very least, this suggests a separate CRT terminal for programmers to use. Depending on the operating software in the supervisory computer, it may be possible for that machine to service programmers during some experiments. It is difficult at this stage to say whether this could be accomplished without impacting the experiment being conducted since the implications of conducting concurrent, unrelated work in a computer depend heavily on the characteristics of its operating system, and since some of these implications can be quite subtle. We would propose first trying to accomplish this in the supervisory computer; if this proves to have undesirable effects on the experiments, a simple expedient would be to provide the software development console with its own LSI-11 microprocessor. This machine would be connected to the supervisory computer, but would need no services from that machine other than access to its large disk or other local peripherals.

OPTIONAL SUPERVISORY COMPUTER PERIPHERALS

The complement of equipment postulated thus far for the supervisory computer constitutes a completely viable system in the sense that each critical task allocated to it can be performed satisfactorily. However, several assumptions are built into that statement. The first is that all programming will be done interactively. While this is not an unreasonable assumption, there do seem to be people in the world who prefer not to adapt to such an environment. Inclusion of a card reader in the configuration may improve the productivity of such persons, but we have not specified one since the investment would be several thousand dollars and would not address

any essential requirement. A card reader might, however, facilitate the receipt of various input data from experimenters.

One of the survey respondents indicated the desire for inclusion of a flexible disk drive. The motivation for this desire lay in the fact that this respondent had a Nova minicomputer which used flexible disks and wished to have a medium whereby data could be exchanged. This is a valid argument for inclusion of flexible disks, particularly since that medium is extremely popular for desk-top microcomputers, and will likely become more popular. We would recommend waiting for a year or two and then considering this medium seriously as a means for getting data into the hands of the researchers in a form they can analyze by themselves.

At various points in the design of MSRF, it was suggested that some provision be made for connecting the MSRF computer system with another for real-time communications. Initially, it was suggested that perhaps the NUC 1110 could "help out" with the system simulations. While conceptually interesting, we seriously doubt that any such help would be of sufficient use to justify the effort of establishing the connection. These doubts arise largely because such external systems are heavily used, and it is difficult to visualize any task the 1110 could perform under such circumstances which would take less time than it would were it assigned instead to the 11/45. However, in connection with experiments where the NPRDC Nova minicomputer was to be used for physiological data collection, it seems essential that an intercomputer communication path be established so that these separate systems may coordinate their efforts. A 9600 baud serial current loop interface is recommended.

Additional serial lines may prove desirable for connection of remote terminals for programming purposes. Support of such terminals, like that of the software development console, depends on the nature of software support actually

implemented in MSRF. We suggest that any decision to provide such interfaces be deferred until it can be determined that their support is feasible.

INTERPROCESSOR COMMUNICATIONS NETWORK

A critical element in the success of the MSRF distributed processing philosophy is the communications link between the supervisory computer and each console. In distributed systems, it can be shown that freedom in allocating functions among the processors increases as a function of the bandwidth of the communications system. We believe that MSRF should implement links capable of transferring data between processors at disk speeds, e.g., 10 to 50 μs per word. The actual speed of transfer should be adjustable to enable "fine-tuning" of the network, since if the speeds are *too* high concurrent use of several links can monopolize memory buses. It is preferable to adjust the system to avoid this problem rather than employing complex software to prevent concurrent use. To minimize communications overhead in the processors, the links should employ Direct Memory Access (DMA) techniques.

Ideally, the links would employ a serial communications discipline, since this simplifies cabling. However, serial interfaces with DMA capability are not presently available for the LSI-11, and, in addition, a transfer rate of even 50 μs per word would require serial bit rates on the order of 320,000 baud in synchronous mode. At present most serial communications devices are limited to 40-50,000 baud, which would severely restrict the bandwidths of the links.

It would appear, at present, that the best choice for interprocessor communications would employ a DR11-B interface in the supervisory computer and a DRV-11B interface in the console computer. Both are available from DEC at a cost of approximately $2,200 per link. Both interfaces provide for DMA transfers, 16 bits wide, at up to memory speed. Some

effort would need to be expended in the solution of any grounding or noise problems which may arise from high-speed parallel communications over long cables, and a small amount of logic would need to be built for synchronizing the two channels and limiting their speeds.

This situation could easily change before the MSRF is built. The systems integration contractor should carefully examine new products, particulary from DEC and from Associated Computer Consultants, Santa Barbara, CA, before selecting the final network interface.

CONSOLE COMPUTER CONFIGURATION

LSI-11 computers are highly modular, and are easily reconfigured (or repaired) by the user in units of these modules. A relatively small quantity of spare PC boards would facilitate rapid correction of problems and would enable simple enhancement of an individual LSI-11 for special purposes. The following module descriptions generally refer to standard DEC components and quantities are scaled for a single console.

Basic Computer

The correct choice of processor module would be the KD11-F ($990) which contains the central processor, I/O bus control, and 4K words of semiconductor memory. It should be acquired with the KEV11 Extended Arithmetic ROM chip ($190), which provides integer multiply/divide and floating-point arithmetic instructions. Owing to the large number of peripheral interfaces, the system should be packaged in an H909-C box ($350) with the DDV11-B backplane ($400). With an appropriate power supply, which cannot be specified until a final configuration has been settled upon, this set of ingredients constitutes the basic computer which may be installed within a console.

Memory

To augment the 4K memory implemented on the processor board, the most cost-effective choice is a 16K-word semiconductor

memory board. Such boards are available from DEC (MSV11-CD, $1,800) or from two other suppliers (with prices ranging as low as $1,000). Selection of a memory board can be done on the basis of price, but it is very important when using LSI-11s in real-time work to include a careful consideration of the methods used for memory refreshment.

A single 16K board would bring the system total to 20K words, which is quite a respectable size. We would recommend beginning with this much and seeing if it is adequate. If not, or if more is needed for a particular application, the machine will accept a maximum of 8K more, which can be acquired from these same sources in units of 4 or 8K.

Real-Time Clock

Although the processor has inherent provision for a 60 Hz clock interrupt, this inflexible (and relatively low) frequency is inadequate for satisfying many timing requirements which we expect will be imposed on the console computers. We would specify the KWV11-A programmable real-time clock ($600), which is capable of measuring intervals of up to 65K ticks, where the tick frequency may be selected to be as high as 1 MHz. This clock includes the vital provision for repeated-interval timing which enables accurate, cumulative timing without errors caused by interrupt latency.

Analog I/O

The need for analog input springs directly from the specification of joysticks and potentiometers as operator control devices, since the natural form of the output from these devices is in the form of voltages. Once the capability to read voltages exists at all, other uses suggest themselves; physical parameters, such as room temperature, can be measured, as can physiological variables such as electrocardiographic or electroencephalographic signals, or galvanic skin response.

Analog output appears, for now, to be less vitally important in the MSRF. The only proposed requirement is to provide signals to control panel meters which may be present in some displays.

There are two sources for analog I/O modules which interface directly to LSI-11s. DEC offers the ADV11-A A/D converter ($1,000), which supports 16 single-ended inputs or 8 differential inputs at a maximum aggregate sampling frequency of 25 KHz and a resolution of 12 bits. The AAV11-A D/A converter module offers four 12-bit outputs. The ADAC Corporation offers a much larger line of products, some of which are less expensive, and which offer higher speeds and more channels.

At this moment the ADAC modules seem to be the most cost-effective. The matter should be investigated again just prior to making an acquisition, but it would appear that any console's requirement for analog I/O interfaces can be met for less than $2,000.

DMA Interfaces

As mentioned previously, the intercomputer link should probably be implemented using the DRV11-B DMA interface ($580). If the Hazeltine system described carlier is chosen to be the primary display system, two more of these interfaces will be required in each console. The DMA interface required to operate the plasma panels with variable button matrices we've specified is included in the price of the panel.

Serial Interfaces

Some keyboards, and some alphanumeric display generators, have EIA RS-232 interfaces. Such devices may be connected to an LSI-11 through the DLV11 interface ($250). These cards are readily available from DEC and should probably be stocked in MSRF in small quantity (one or two) against their need.

Parallel Interfaces

Certain other keyboards and devices have parallel TTL interfaces which may be dealt with by the DRV11 parallel interface card ($210). The need for these cards should be determined when the consoles are constructed.

Generalized Interfacing

One source of extreme nuisance which can easily arise in systems such as MSRF involves the technical difficulties which can attend such conceptually simple acts as interfacing the computer to a single new switch. Indeed, these difficulties are present to one degree or another with most discrete display and control elements. We feel that this problem area can be helped significantly by the expenditure of some forethought and design effort as the consoles are being constructed.

The electrical characteristics of the analog signals produced and expected by the discrete elements which may be mounted in a console are often incompatible with those respectively expected and produced by A/D or D/A modules. Usually, some sort of buffer amplifiers must be employed to reconcile these differences. Several manufacturers, such as Heath-Schlumberger, make excellent lines of modular analog processing modules which can aid in solving these problems without promoting dependence on locally fabricated circuits for most purposes. We would recommend the inclusion of a rack for such modules, and the inclusion of any modules suitable for interfacing to the discrete elements which are actually procured for MSRF. Others could be acquired as the need arose.

Unfortunately, there does not seem to exist any similarly rich line of modules for processing digital signals, such as switch closures. We would recommend that NPRDC fund the development of a general-purpose, modular interfacing system for such signals. We would envision this system to be capable of handling roughly twice the number of discrete devices initially

conceived to ever be installed on a console. The system would need to offer three levels of interface:

1. Signal-conditioning modules, for adapting to differing electrical characteristics of devices and for performing such functions as switch debouncing.

2. Logical interface modules, principally for input processing, which would provide for matrix-coding of buttons or for generating interrupts when the states of switches were changed.

3. Computer interfacing, which would generate interrupts and make available to the processor a set of registers representing the states of input and output devices.

A rough block diagram of the generalized interfacing subsystem appears in Figure 15. Interconnection and configuration of the subsystem should be flexible and should employ patch-panels, matrix switches, or other devices which facilitate the setup of an experiment. A part of this development effort would be the specification of the electrical characteristics of those devices to which it could interface, and a description of the procedures for interfacing incompatible devices.

Power Supplies & Cabling

Several other physical elements are called for in the consoles. DC power supplies will need to be present, apart from those in the console computer, to provide voltages required by the various discrete modules. Specification of these supplies' outputs should be done after the repertoire of modules has been selected, and should provide ample spare current for future devices. We anticipate that the required supplies should cost less than $2,000. In addition, each console will require a power distribution and circuit breaker panel. The need for blowers can be avoided by leaving the tops and bottoms of the consoles open and by providing for

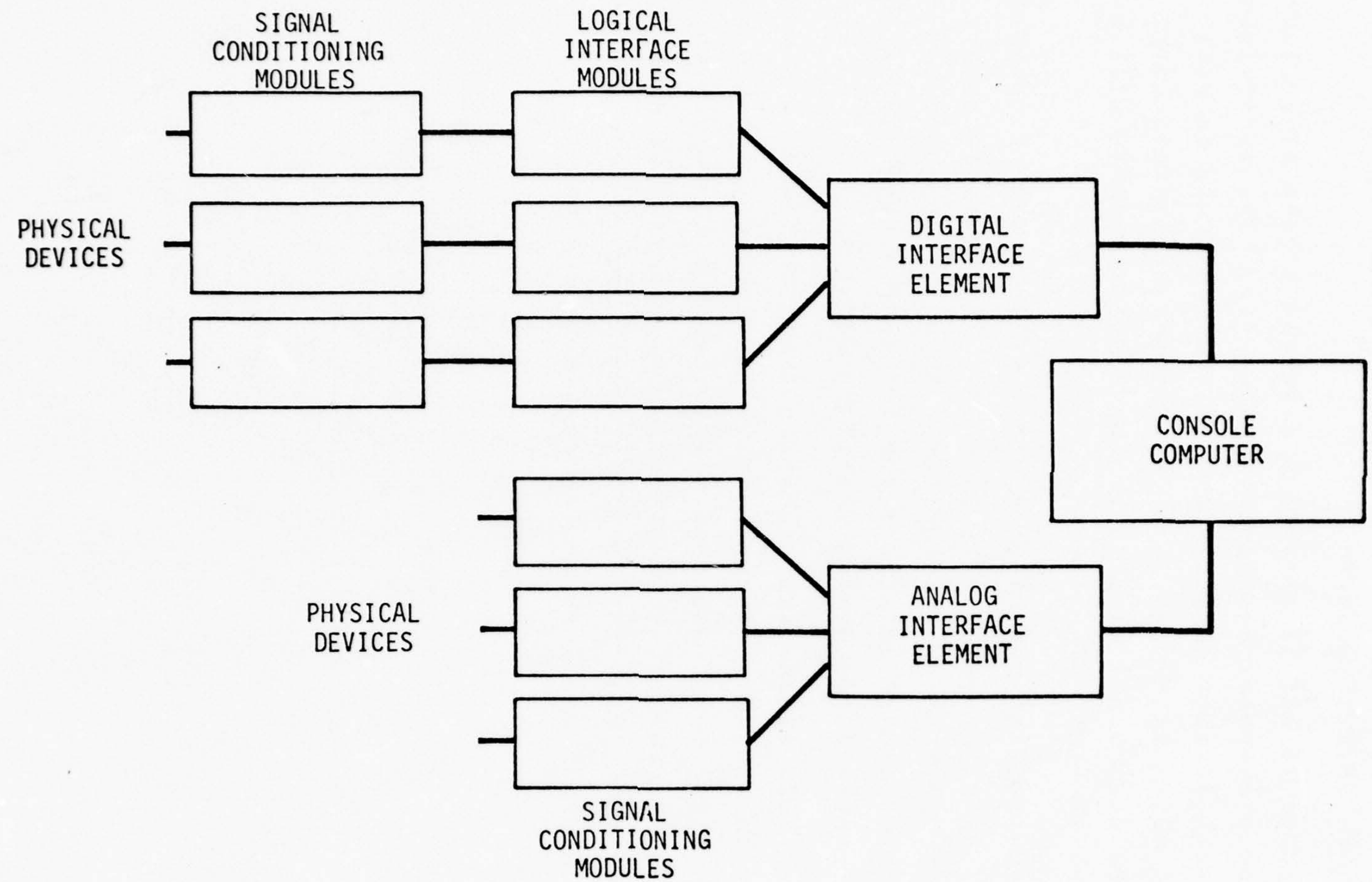


Figure 15. Generalized interfacing subsystem.

flow of conditioned air upward through the consoles from beneath the computer floor.

SUMMARY OF COMPUTER SYSTEM

Figures 16 and 17 are block diagrams of the supervisory and console computer systems, respectively. The supervisory system will consist entirely of off-the-shelf components; which of these will need to be acquired is uncertain until it can be determined exactly how much hardware ATTS will be releasing to MSRF.

Likewise, the console computers will be almost entirely composed of off-the-shelf components. The only two areas which, at the time of this writing, appear to require design or development work are the interprocessor network interfaces and the generalized digital interface modules.

We should once again emphasize that the field of products relevant to the MSRF computer system is subject to rapid and dramatic change. It would be highly prudent to survey this field again prior to final selection or ordering of components for MSRF, with the possible results of reduced cost or increased employment of readily available equipment.

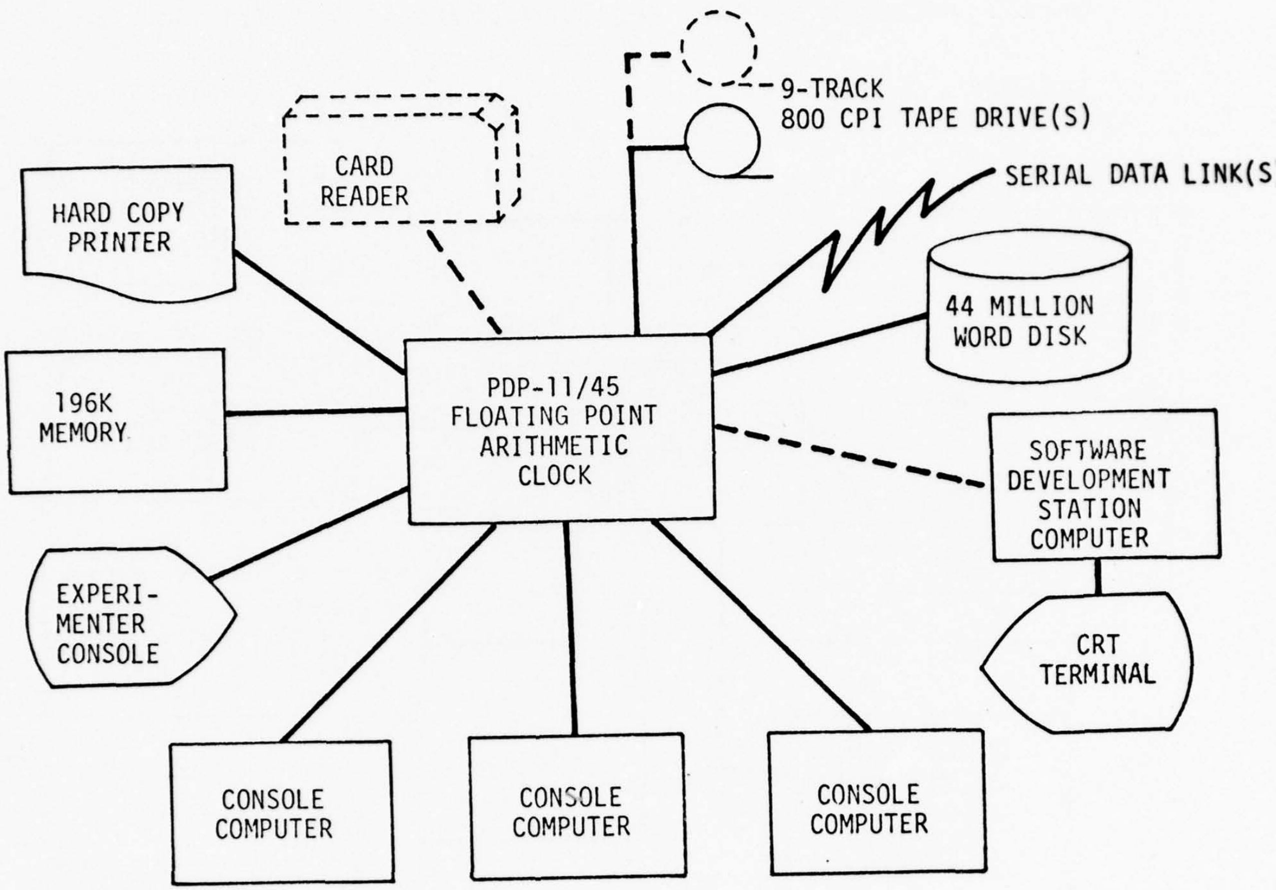


Figure 16. Block diagram of the supervisory computer system.

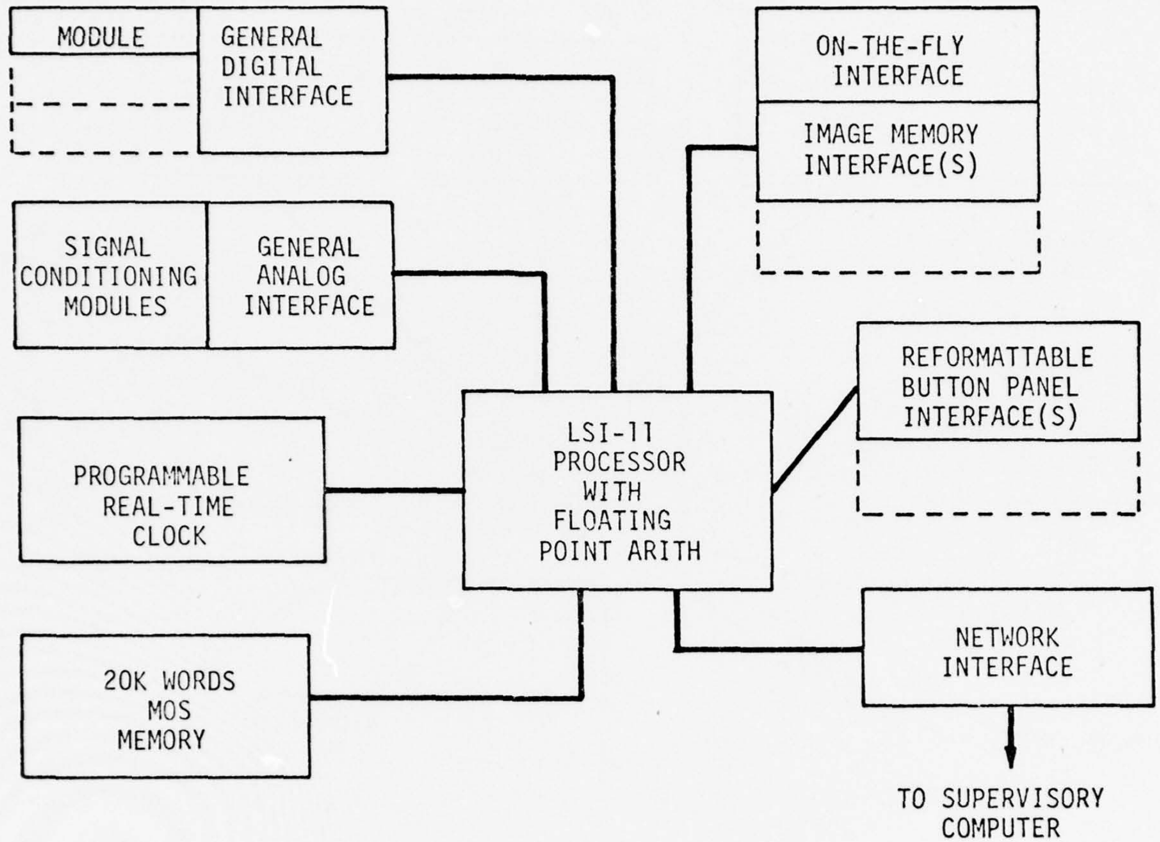


Figure 17. Block diagram of the console computer system.

SOFTWARE REQUIREMENTS

Preliminary Discussion

The importance of software in the success of the proposed design for MSRF is paramount. The facility derives its flexibility by assigning total responsibility for its behavior to the realm of computer programming. The responsibilities which accompany that assignment cannot be overemphasized.

For each experiment to be conducted in the facility, there will exist a corresponding body of software responsible for controlling the experiment, defining and implementing the behavior of the system under study, and collecting data of interest to the researcher. The demands placed on these bodies of software will vary widely from experiment to experiment. While simple experiments, dealing with simple systems, can be dealt with by trivial programs, there will undoubtedly exist experiments calling for extreme complexity, extreme speed, or a combination thereof. Regardless of the complexity of the programming required, it must be recognized that *some* level of software effort will be called for in setting the laboratory up for each experiment.

Since no experiment can be conducted without some sort of a software effort, and since history has taught us that such efforts can be expensive, time-consuming, or unfruitful if gone about incorrectly, it would seem relevant to explore the personal and environmental attributes which are critical to inexpensive, timely, and successful programming work.

It should be obvious that familiarity with the equipment in the facility is essential to efficent work. Those writing software for experiments must understand the behavior, capabilities, and limitations of the hardware, and what is required to exercise these capabilities before they can accomplish

useful work. Shallow or incomplete understanding will result in inaccurate assumptions; while many such assumptions would lead only to minor errors which could easily be corrected, they can often lead to gross design flaws which can only be corrected by junking large amounts of software when they are finally discovered. It is often claimed of utility software packages that they reduce the need for programmers' familiarity with the hardware; while occasionally this is true, their typical effect is to reduce the flexibility of the hardware and, by concealing its nature, to promote those very shallow understandings which lead to misuse of the hardware. Since this familiarity with the equipment is necessary, any research project whose programming is to be done by people unfamiliar with the facility must be prepared to pay the price of their education and of their mistakes.

Another important body of knowledge for an MSRF programmer is familiarity with the Naval systems to be studied. Acquisition and maintenance of this body of knowledge can be more difficult than meets the eye, especially in the case of those systems that are themselves computerized. The behavior of such systems as NTDS is continually changing; documentation of that behavior has, in our experience, been difficult to come by and voluminous once one has found it. While we are asserting that the MSRF, because of its software-intensiveness, *can* faithfully simulate a given system's behavior, the assertion is only valid when that behavior is adequately defined. In some cases, the necessary understanding of a given system can be acquired only by scanning hundreds of pages of documentation and asking hundreds of questions; and the understanding can be maintained only by receiving and studying notification of changes and/or by maintaining personal "hands-on" familiarity with the current operational equipment. Naturally there will exist experimental designs which can produce transferable results without demanding precise fidelity. On the other hand, function and system behaviors at the man-machine interface

are central issues in much of the proposed research, and should presumably be implemented faithfully. While the maintenance of adequate documentation of the target systems will help programmers to discover what they need to do, it is evident that they will require considerable time to assimilate and refine this information, particularly since most of the documentation is written for maintenance training purposes and fails to answer many relevant man-machine interface questions.

Computer programming is a task requiring the exercise of creativity. Although the degree to which this attribute is required depends on the problem to be solved, our understanding of the complexity of such systems as NTDS and Naval sonars is such that we expect that their simulation for meaningful research will require creativity in large quantity. The world of real-time simulation programming is not a simple one, nor are its natural laws easily taught. It is a very practical world, one which embarassingly punishes those who are naive enough to underestimate it. Few other areas of computer applications so thoroughly tax both hardware and software, principally because all work to be done by the system must be accomplished on a fixed time schedule. Since time is the one resource which is not critical (except economically), in most other application areas different attitudes are required. The accuracy of calculations must be traded off against the time required to perform them, and one must understand how the resulting inaccuracies propagate and control them. The organization of data bases must in many cases be entirely different than they would be for non real-time applications. In an environment such as MSRF, where multiple computers are involved, the successful implementation of complex simulations may require unusual allocations of workload among computers, novel managerial structures within the system, and even new conceptual tools. In many cases, a problem may only be solved by completely restating it. These are not common skills, and often they imply knowledge which crosses many disciplines.

The preceding discussion leads us to several definite conclusions:

1. That the equipment in MSRF must be well documented.

2. That NPRDC must acquire and maintain adequate documentation pertaining to the Naval systems of interest.

3. That permanent, key programming personnel must be employed by NPRDC who will become familiar with both MSRF equipment and with the systems to be studied, and who are given time to maintain that familiarity.

4. That it would be foolish to predicate the success of a research project on the capabilities of individuals who lack knowledge of the MSRF equipment, the systems to be simulated, or the disciplines required to simulate them.

In specifying the software support for MSRF, we will assume that competent personnel will be doing the majority of the programming and that they will have adequate knowledge of the MSRF hardware and the nature of the systems to be simulated. The problem is then one of defining tools which will be of use to those personnel. There are two radically different ways in which this definition may be undertaken:

1. Try to anticipate every problem which will confront the programmer. Construct a modular routine to solve each such problem, and attempt to integrate these routines into a coherent operating system.

2. Try to identify those tools which will be useful in the solution of every problem. Organize from these an operating system whose modular components may be deleted, if unnecessary, or augmented by new components as the need arises in a particular application. Make this "tailoring" function as simple as possible.

The first method leads to what we shall call "traditional operating systems," which will be discussed next. Subsequently, we shall describe a recommended alternative.

Traditional Operating Systems

The first method of defining programming tools is by far the most commonly used, largely because it is historically the way in which software tools have grown into being. As the art of programming matured, people became aware that there were many common problems needing to be solved over and over again. Some of these, such as calculating approximate values for transcendental functions or communicating with peripheral devices, were sufficiently complex that their continual re-invention contributed significantly to the effort of writing and debugging each application. This situation was alleviated by creating the necessary tools to support modular programming, and by creating "standard" modules for performing those functions commonly needed by application programs. As time progressed, great volumes of these modules came into being, and it became necessary to exert some managerial control by organizing them into libraries. Duplications were weeded out, the modules were rigorously tested and documented, and stringent controls were imposed upon their alteration. This last point is a significant one. Once a program module has been placed in a library and used in numerous applications, that module's behavior cannot be changed in the future in such a way that easier applications which depend on its original behavior will cease to work.

Presently, such a modular library exists for virtually every computer made. These libraries contain routines for doing anything which has been deemed to be commonly useful. For the sake of economy, these modules use each other where-ever possible, so that changes made to the behavior of one module often necessitate changes in others. Taken together, such libraries are tightly woven structures which are highly complex and difficult to change in any significant fashion. Taken together, such libraries also impose tight constraints on the form and function of application programs which depend

on them. Such constraints are tolerable, and perhaps desirable, for many applications; the inefficiencies imposed are relatively invisible, since most applications are batch jobs whose usage of resources is relatively insignificant when compared to the accuracy of their output.

For real-time applications such libraries or operating systems are often of negative value. Taking a trivial example, the usual criterion for selecting a given approximation of a transcendental function is its accuracy. While such common library components as square root, log, exponential, or trigonometric functions are normally optimized for speed, this speed is usually limited by the typical demand that they produce results as accurate as can be represented in a data item of some type. In a batch environment, any less will result in endless complaints from users. In a real-time environment, no algorithm is of any use at all unless it produces results in a sufficiently short time, regardless of its accuracy or lack of same. This is true as well of every other module whose services a real-time application may invoke.

This leads us to an interesting essential difference between the processes of developing real-time simulation software and other types. For normal software, one fixes the accuracy and method of a program, and lets its usage of time vary commensurately. For a real-time program, the usage of time is fixed and it is instead method and accuracy which must be adjusted. The more such a program depends on an immutable operating system and library software designed for accuracy, the less its chances will be of achieving a solution at all in a demanding environment.

Many computer manufacturers offer what they claim are real-time operating systems, even though these systems possess the defect of immutability. This enables them to sell more hardware; when an application cannot perform in real time due to the constraints imposed by the operating software, the suggested solution is to add more memory or more

processors rather than risking any change to a complex operating software package. Unfortunately, most modern operating systems are of sufficient complexity that significant changes to their behavior *are* beyond users' capabilities, and for these who depend on such systems, hardware modifications are often the only viable recourse.

Another point of interest regarding traditional operating systems is brought forth by the fact that MSRF may properly be described as a distributed processing system. It employs several computers which act in concert, sharing the processing burdens imposed by a given experiment. This is emphatically *not* a new concept. Distributed processing has been used to solve special problems for well over a decade. What *is* new about distributed processing is that within the past 2 years or so the computer manufacturers have been pushing to place the technique within the grasp of the average applications programmer. Naturally, the means whereby they propose to accomplish this is to extend the structure of traditional operating systems to encompass multicomputer environments, a solution which *will* be replete with the aforementioned attributes of immutability and of imposing restrictions on the form and function of applications. A key word here is "will"; to support distributed processing in a traditional way, a "general" solution must be found. Since there are even more potentially good ways of organizing several computers than there are of organizing a single machine, it is not surprising that debate over the "best" distributed processing system rages on, and that the concept has not yet been implemented for many machines. When, or if, some sort of agreement is reached, we may trust that the general solution will be fine for distributed accounting and will be of marginal usefulness in meeting the needs of real-time simulation applications.

Were we to recommend that MSRF employ a traditional operating system as the principal tool for aiding in software development, we would offer four alternatives:

1. Purchase from DEC its currently most advanced "real-time" system, along with DECNET which is their current mechanism for interprocessor communication.

2. Purchase the UNIX operating system, acquire from UCSD its modifications for interprocessor support, and integrate these together.

3. Identify and procure some other real-time distributed operating system which may exist but of which we are not aware.

4. Define a totally new traditional operating system and fund its creation, expecting the cost to be on the order of $100-200 K.

Were any of the first three paths to be chosen, we would recommend that MSRF procure the complete source for the selected operating system and whatever tools may be required to enable its complete local maintenance.

The implications of choosing any of the first three paths are classic. One may expect to find bugs in the system he has acquired, and should expect to correct them himself or else wait weeks, months, or years for their correction by the vendor of the software. For real-time simulation work, one should also expect that he will need to make modifications to the system. What MSRF would be getting itself into would be the need to fund a software maintenance contract, to train and maintain an in-house systems programmer who is familiar with and capable of modifying the entire system, to maintain an internal set of corrections and alterations to the system, and to retrofit these corrections and alterations into each new version of the system received from the vendor. For a body of software consisting, typically, of 100 to 300 thousand lines of code, the cost and nuisance of these things can be astounding.

The fourth path would lead to a somewhat simpler future in that all future development would be done in-house, alleviating the retrofitting task. However, while we have designed many operating systems in the past, it is significant that

we would not ourselves hazard to define such a system since we feel that traditional operating systems *cannot* solve the problem of maximizing what can be achieved with a given hardware system.

In fact, we do not recommend use of traditional operating systems to support MSRF software efforts. It has been our experience that, for real-time simulation applications, such systems create more problems than they solve; that the required performance can often be achieved only by subverting such systems; and that often the required subversions may be achieved only by dispensing with the systems entirely.

A Recommended Alternative

The second general way to go about designing a set of software development tools requires far less work and results in a far less massive body of programming. This is true largely because it is accepted from the outset that the needs of real-time applications vary widely, that their problems will differ, and that no single structure will satisfy the needs of all applications. Consequently, systems defined according to this method impose very few absolutes on those who use them, have highly mutable structures, and are in fact quite simple. Systems of this type often leave systems designers who never implement applications quite unimpressed. On the other hand, systems of this type are, in our experience, of great practical use to those who ultimately have to contend with systems--the application developers.

There exist several bodies of software of this general type. One, called FORTH, was first described in the literature by C. H. Moore in 1974,[1] and later in 1976.[2] This package is

---

[1] Moore, C. H. FORTH: A new way to program a mini-computer. *Astron. Astrophys. Suppl.*, 1974, *15*, 497-511.

[2] Rather, E. D., & Moore, C. H. *The FORTH approach to operating systems.* Paper presented at the Proceedings of the Annual Conference Association for Computing Machinery, Inc., 1976.

currently available from FORTH, Inc., Manhattan Beach, CA. It has been implemented on over 20 different types of computers and has been used as the foundation for hundreds of applications. HFR and Athena Programming have developed a proprietary package called FLUX which is very similar to and compatible with FORTH; this package has been implemented on three types of computers and has been the implementation vehicle for all new applications software written at HFR in the past year. Other software packages of similar design are appearing at various places as people become aware of their advantages.

Simply stated, the underlying principle of FORTH-like systems is that the primary problem in computer programming is the need to define procedures to be performed by a computer. All other problems, such as data manipulation, file management, and arithmetic are in fact higher level considerations and are seldom the same for any two programs in either nature or importance. Therefore, no presumption is made about the means to be employed in solving even these "basic" problems.

In terms of traditional software systems, this is a highly revolutionary concept. An application which merely watches for a given event to occur, and turns on a light when it does, is so beneath the dignity of modern operating systems that it would not be supported well at all. While such a procedure could be implemented, in its entirety, by a mere handful of instructions, its execution in a "normal" environment would still require the presence in memory of thousands of words of operating system, *none* of which would be of any use to the application. The alternative to this absurd overhead would be to construct a "stand-alone" program; some operating systems *do not even support* the construction of such programs, and those which do force the developer to use only the most primitive of tools in their implementation.

FORTH-like systems are built in "layers." Each conceptual layer, or shell, depends on the mechanisms provided

by what it surrounds and provides new, higher-level mechanisms for use by outer layers. The innermost core of a FORTH-like system is the hardware; the outermost is usually the operator command level, at which administrative interaction occurs with the application. Significantly, each layer, while providing higher-level or more application-specific ways to use the capabilities of the inner layers, does not forbid access to the inner layers. Thus, the hardware base, and everything in between, remains directly and simply accessible from even the outermost layer in an application. (This access, however, may easily be denied by programming to create a "sailor-proof" system.)

An application is constructed by selecting or writing appropriate layers of software and by using these to build a system. Only those layers, or parts thereof, which are actually needed by an application must be included. The result is very compact applications which may run on smaller machines, or accomplish more in large machines, than may applications which are constructed otherwise.

One relatively standard application of FORTH-like systems is the construction of software. Like all others, this application is formed in layers, and may be extended or altered as desired for a particular site, or even for a particular application. A typical software construction environment contains the following basic features:

1. A macro assembler which may be used to generate machine language instructions or other data from symbolic text.

2. A compiler for a fully user-extensible high-level programming language with a high-level macro facility which generates extremely efficient and compact interpretive code from symbolic text.

3. A convenient basic convention for the organization and formatting of source programs on mass storage.

4. A text editor for interactive writing and updating of source programs from terminals, with

utilities for manipulating mass storage contents, listing programs, and performing other typical maintenance tasks.

All of the above processors and utilities employ a common set of language rules which may be extended or altered by the programmer. Virtually no limitations are imposed on program structure, although a simple set of conventions is available for programmers' use if appropriate. The tools normally supplied above assemble or compile programs *interpretively*; thus, source programs are not merely symbolic data but are in fact *programs which are executed by the construction applications*. The result of executing such a "program" is the generation of code. This results in an environment where there exists no meaningful distinction between "job control language" and "programming languages."

The basic software development application is usually reentrant and may be configured to suit the situation at hand. It may be a part of a dedicated, single-user system, or it may be used to construct a multi-user development environment. The application may be used to compile programs for immediate execution and test in the development environment, or cross-compilations or assemblies may be done to generate tailored stand-alone environments or even software for different computers. Programs may be constructed in either machine or high-level languages exclusively or in any combination leading to a satisfactory problem solution.

The environment is particularly well-suited to interactive program checkout. Primitive functions of the application, even single machine language instructions, may be exercised directly from a terminal and their behavior observed with very little effort. This capability is of great value as well in the exploration of new equipment's behavior or in diagnosis of hardware malfunction--ordinarily very difficult areas to deal with. We consider interactive checkout to be especially important in applications dealing with computer graphics, a particularly relevant consideration for MSRF.

The software development environment can be easily tailored for a site's requirements. In MSRF, for example, it would probably be worthwhile to add word processing and other utilities to aid in the production of documentation. Various special tools for aiding in program debugging or performance analysis can be provided in a general form so that they can easily be adapted by a programmer to his particular problem.

What is involved in fabricating a workable system environment for an application depends, not surprisingly, on the nature of the application itself. As we have said, the programmer has assembly and high-level languages to use as he sees fit. Any combination of the two may be used to construct a program which runs in either the development environment or in a stand-alone mode. The former is convenient during checkout if the space occupied by the compiler, assembler, and text editor can be afforded (usually less than 4,000 words) since these tools may be used for all sorts of diagnostic purposes, including dynamic monitoring of programs and data bases during operation. In fact, this latter capability could easily form the nucleus of a mechanism for extracting experimental measures. If the space for these tools cannot be tolerated, then they can be dispensed with. Yet this decision is up to the developer of a given application; simple problems may be solved in a simple and straightforward manner, rather than being *forced* to contend with complexities merely because some other application requires them.

Multiprogramming, reentrant coding, intertask communication and synchronization, dynamic resource allocation, and other esoteric concepts are frequently required in real-time applications. Yet there are many ways of implementing each of these, and each implementation method has its own set of implications with respect to constraints placed on the programmer, space and time overhead, and ultimate system behavior in such terms

as responsiveness and overloadability. Multiprogramming may, for example, be planned or interrupt-driven, time-sliced or prioritized, or of fixed structure or having dynamic priority rearrangement based on program behavior. Intertask synchronization, may be achieved by direct stimulation, event control blocks, or message queues. These lists are highly incomplete, but each term implies a highly different system organization and behavior. Conventional systems are so organized as to support one combination of these methods, and to change the method used is normally an undertaking requiring major surgery. Yet each method has definite advantages and disadvantages in the context of a particular real-time application. Those applications which are ill-suited to the methods forced on them by a particular system can be extremely difficult, if not impractical, to implement on that system.

For this reason, FORTH-like systems do not assume any particular choice of multiprogramming methods, and instead permit the application developer to choose one which is well-suited to his problem. For MSRF, examples of several methods should be constructed to form a *library* which may be used as is or modified to taste by the programmer.

This same philosophy can be extended to other areas as well. There could be several alternative disciplines for interprocessor communication, offering alternative trade-off points between speed and generality. The same is true for methods of constructing and loading programs into the console microcomputers, and for managing the various console peripherals. In contrast to a conventional, inflexible operating system, a FORTH-like operating software base for MSRF could consist of a library of user-tailorable functional modules which could be combined to form a variety of radically different operating environments.

The same is true of file structures and internal data representations. Conventional programming languages presume

to define a limited set of ways in which data may be represented in a program. For example, most FORTRANs support only integer, real, logical, and complex variables and arrays. If an application needs to deal with, say, fixed-point fractions, extremely large integers, vectors, or matrices, these things may be accomplished with suitable machine-language coding, but the programs which result become cumbersome, unreadable, and, worse, are often highly inefficient. FORTH-like high-level languages permit the user to define and operate on any type of data he can describe in a convenient, readable, and *efficient* manner. File structures are likewise programmer-definable. For real-time applications, specialized data types having reduced precision and special types of arithmetic are often required in order to solve the problem, and such may easily be defined. As with the "operating system," MSRF could have a library of definitions for various types of data and arithmetic which could be used as appropriate.

One way to illustrate this software concept is to cover the high points of the design of an application for a simple experiment not unlike one which might be conducted in MSRF. Let us suppose that a researcher is interested in studying operator performance variables in a system like NTDS, and has arrived through the results of a task analysis at the conclusion that ball-tabbing of objects on the display is a critical task for certain operators. He has concluded that these operators' performance depends heavily on the speed and accuracy with which the operator can designate a given object. This leads him to a set of hypotheses having to do with the effects on performance of differing relationships between trackball movement and cursor response on the displays. He designs a program of experimentation to test these hypotheses and wishes to utilize the MSRF.

Let us further suppose that up to three subjects at a time will be sitting at simulated NTDS consoles. Each subject

will receive a series of problem presentations on a fixed schedule. Each presentation will give him a static, preselected display of one or more NTDS symbols, one of which is of a specific type. His task is to manipulate the trackball to place the display cursor sufficiently near this object to identify it unambiguously and to then press his "hook" button. Each presentation will employ a predetermined algorithm "connecting" the trackball with the cursor. The data to be collected for each presentation are: whether the "hook" button was pressed within the time limit, and, if so, when it was depressed relative to presentation of the display; and the position of the cursor when the button was pressed. Times should be accurate to 1/60 second, and position accurate to within the smallest unit displayable. Later analysis will determine how far the cursor was from the target object, and whether this position enabled a successful "hook."

Whether or not this would be a good experiment, it would be easy enough to implement in the MSRF environment we postulate. To begin with, the central problem to be solved is implementation of the various cursor behaviors. Let us suppose that these are expressed as either servo systems or differential equations, with or without simple logic for speed breakpoints. The problem is, in either case, to incrementally exercise the servo or integrate the differential equation at a fixed frequency.

From prior study of the trackball, we might have concluded that its inertial low-pass filtering of manual inputs guarantees that frequency components in its movement in excess of 50 Hz are sufficiently attenuated that they are negligible. Therefore, a sampling frequency of 200 Hz will prevent aliasing, and, since it is higher than the frame-refresh rate for the video displays, should not contribute to display jerkiness. This is viable since it gives us 5 ms of computing time for each cycle through the algorithm, and our experimenter has

not defined any functions too complex to update in 1 or 2 milliseconds.

The obvious place for this simulation to occur is within each console microcomputer. At the most primitive level, the trackball apparatus contains a register which is counted up or down by one for each 1/1000 revolution of the ball in X and Y directions, and this counter may be interrogated or reset by the microcomputer at any time. The major cycle of the simulation will consist of reading and resetting these values every 5 ms, inputting these to the selected algorithm as deltas, evaluating the functions, and outputting the newly calculated cursor position to the display.

Assumedly, coding to interrogate the trackball already exists, but if not it can be written in minutes. We remove from the library the basic graphics package for the primary display, along with graphics routines for drawing NTDS symbols and a cursor of the type we want, and the routines for delaying until a specific time. The implementation may begin by collecting these things together and by writing one or two of the experimenter's algorithms in the high-level language.

Let us assume that the normal program development package includes copies of the compiler, assembler, text editor, etc., in each of the microcomputers as well as the supervisory computer, and the ability to connect various terminals to various machines over the interprocessor link. We seat ourselves at one of the subject consoles which has an ASCII keyboard, connect ourselves to its microcomputer's development software, and proceed to load the aforementioned utility routines and compose one of the trackball algorithms. We invoke it, manipulate the ball, watch the cursor, and see if it seems reasonable. Perhaps, if the algorithm in question is supposed to have a particular transfer function, we disconnect the algorithm from the real devices, load trigonometric and graphics routines, and stimulate it artificially to obtain a frequency

response curve. If we have hard-copy graphics capabilities, perhaps we output these curves so the experimenter can include them in his report.

The foregoing paragraph may sound like a fantasy to many applications programmers. In fact, it has been our experience with FORTH-like systems that this sort of interactive checkout could easily be done for *several* algorithms in a relatively unhurried afternoon's work.

Having written, tested, and documented the trackball algorithms, we really have very little more to do. By pulling in the simplest file package available, we define two files--that which contains the sequence of stimuli, and that which will receive the data extracted from the experiment. Again working in the high-level language, we write the control loop which will, on a fixed time-cycle, extract an item from the stimulus file, display the specified targets, initialize and connect the specified trackball algorithm, and then, on a 5 ms schedule, cycle the algorithm, check the status of the hook button, and repeat until either the button is pressed or the subject's time is up. We then write the relevant data to the output file and begin a new cycle or stop if the experiment is over. A small amount of coding within the supervisory computer will enable us to log a given subject in and begin his sequence of presentations. After receiving the specifications for the stimulus sequence from the experimenter on some medium and entering them into the sequence file, we need only provide for writing the output files from the experiment onto magnetic tape in whatever format is suitable for the data analysis software, and the programming task for this experiment is complete.

This overview has been very general, yet it suggests quite accurately the typical path of progress of application programming in FORTH-like environments. Software is written in small pieces which are readily tested. An application as

simple as this one could run in the program development environment (because it is small and does not call for special multiprogramming support), could be quite comfortably implemented within about a week if most of the tools we postulated were available, and would likely require only several pages of new software. Large-scale NTDS or sonar system simulations would require substantially more planning, effort, and time, and would likely present problems requiring careful thought. Yet the general process would be quite similar.

All of the foregoing may seem to claim that everything about FORTH-like systems is good. Naturally, this is not true; as is the case with all other systems, these have their own sets of disadvantages. As we see it, it is the very flexibility and power of these systems which leads to most of the complaints which people who have used them can think of. FORTH-like systems generally produce few diagnostics. If one writes statements whose meaning is unclear or whose interpretation logically leads to a system "crash," he may expect ambiguous results or the need to reload the system. Traditional systems *attempt* to examine statements or programs for inconsistencies, and in some cases glean small parts of their meanings. If the results of these examinations fail to pass certain tests, then the programmer is informed and often the system will refuse to execute the program. Some people consider this sort of diagnostic activity to be essential. We do not think it is that important, for several reasons:

1. The true intelligence embodied in these tests is infinitesimal, and they will usually detect only trivial errors. Programs which pass all the tests will still often not work properly, for far less trivial reasons. The patterns detected by traditional diagnostic coding are not selected on the basis of their importance or drastic implications for the application, or even the frequency with which the error is made. Instead they are chosen on the basis of the ease with which they can be recognized.

2. The frequency of trivial errors diminishes as one acquires fluency in a language, resulting in a parallel decrease in one's dependence on trivial diagnostics. Moreover, the interactive debugging capability of FORTH-like systems leads to immediate detection and elimination of these errors as part of the normal sequence of writing programs.

3. It is demonstrated very effectively by comparing the size and complexity of FORTH-like and traditional systems that diagnostic capabilities contribute significantly to both of these areas. Diagnostics add overhead, increase maintenance burdens, and contribute to the inflexibility of a system.

This, and some less significant disadvantages, definitely accompany the choice of a FORTH-like system. We feel on both theoretical and practical grounds that they are by far outweighed by the advantages of these systems; for, after all, one must ultimately implement his application, and it is the solution of this problem to which FORTH-like systems offer unprecedented assistance, both in initial construction and in the detection and analysis of those design problems which would "slip by" traditional diagnostics anyway.

We heartily recommend that MSRF employ a FORTH-like approach to its software support environment, as the most practicable we are aware of for aiding in real-time simulations. It must be remembered at all times that the responsibility of MSRF is to conduct experiments. No "software requirement" may be considered relevant unless it contributes directly to the fulfillment of this responsibility, and certainly not if the requirement in any way *detracts* from the conduct of experiments. Real-time simulation is difficult enough by itself that the facility can countenance no artificial constraints on its software support which might have been stipulated in a "traditional" system to accommodate the needs of applications almost totally unrelated to those of the Manned System Research Facility.

Specific Requirements

Regardless of the method chosen for creation of the MSRF software support, it is possible to list a set of requirements. Each item listed is one which we feel should receive attention and which should be provided for with appropriate algorithms. Several of these should properly be supported by several alternative algorithms.

PROGRAMMING LANGUAGE PROCESSORS

Assembler

There must be a processor enabling the programmer to work in machine language. It should have a good macro capability and should enable the programmer to generate coding which exercises any part of the hardware. The assembler should be so integrated with other language processors that the programmer may easily use a combination of assembler and high-level languages in the construction of a program. It should offer means for constructing arbitrary data structures as well as machine language instructions.

High-Level Language

Some sort of high level programming language should be available. The language should be one which is especially useful in writing heavily procedural programs, and should not penalize the user unduly in space or time for invoking subroutines or subprocedures. The language chosen should make it easy to write about what is going on in MSRF, meaning that it should be capable of adapting linguistically to the multitask, multicomputer environment of the facility. It should, likewise, be capable of adapting to special data- or program-structural requirements of a given application. It should enable intimate contact between the hardware and the high-level programmer. There should exist a high-level macro capability. It should be easy to write reentrant programs.

All programming languages to be used in MSRF should easily support interactive programming and checkout. This implies interactive, symbolic debugging capabilities from consoles, and a viable text editor. The languages should enable programs to be written concisely since lengthy programs are not easy to work on interactively. Batch-type programming is definitely *not* recommended for MSRF.

The object code for the high-level language should be inherently compact, and it should be made possible for the programmer to produce code representing the extremes in any trade off between space and speed, as he sees fit. It is of paramount importance that compactness be achievable; "overlay" programs are seldom desirable for real-time use, so complex programs must be compactible to fit into available memory in their entirety.

Problem-Oriented Language Metacompiler

There must be a tool enabling the application programmer to create languages for use by personnel (e.g., researchers) who are not experienced programmers. These languages would be used for overall control of experiments and for the definition of data extraction procedures. While requirements in this area will be highly variable, the tools must be easy for the application programmer to use and must result in languages which are easy to use.

All programming languages selected for MSRF should be subjected to the tests implied above. They should lend themselves to rapid learning at the level of the student, meaning that one who understands computers well should be able to rapidly grasp the entire environment, while a dilettante who only understands a few of the underlying concepts should be able to rapidly acquire the degree of fluency required to express those concepts. In any contest between diagnostic capability, or restrictiveness, and power of expression, the winning language for MSRF should be that which enables a person with deep thoughts to express himself well rather

than one whose restrictions render it difficult to make a mistake while making it difficult to say anything of significance.

OPERATING SYSTEM FEATURES

We feel that MSRF should incorporate at least three distinct "operating environments."

The first would be that normally used for program development and checkout. It would place all programming tools, all processors, and all peripheral equipment at the fingertips of the user. This would be a multiprocessor, multiprogramming environment which could be reconfigured within some reasonable set of constraints, and within which a subset of experiments could probably be run.

The second would be a completely flexible architecture whose construction from standard or nonstandard components places all system resources under the complete control of the application programmer.

The third would be a modular environment which could be selectively included as a part of either of the first two. It would offer many of the facilities of the development environment to multiple remote users for program development and checkout; however, hardware and software restraints would be employed to prevent such users from disturbing or affecting the principal application in progress.

The third environment, while desirable, represents a completely secondary need. It should not be implemented unless the precondition of nondisturbance can be met, and it should only be included in a given environment if this can be accomplished without forcing any visible sacrifice on the application.

The order in which relevant operating system requirements are listed below should not be taken to imply any particular hierarchy, since such hierarchies vary widely across system designs.

Hardware Management

Each peripheral device in the system should be studied carefully to determine the requirements for interfacing with it. Sometimes such studies result in the realization of timing dependencies, ambiguous failure modes, or interdependence with other peripherals. Efficient device drivers should be written which solve this interfacing problem without presupposing the structure of data interchanged with the devices. Error recovery should be taken to lengths appropriate to the probability and consequences of an error.

Other hardware mechanisms, such as memory mapping or protection, alternate register sets or computer states, and real-time clocks, should be considered. Basic mechanisms to use these things should be present, but hopefully the operating system should avoid depending on them itself. These can be very powerful resources in real-time work if they are made available to the programmer.

Data Management

MSRF operations, while not oriented toward data processing as such, will still require tools aiding in the management of data. Online storage will be used as a repository for programs and for data files representing the input to and output from experiments. Tools must be provided for managing the allocation of mass storage for these files, and others must exist for describing their contents to programs using them. While various file structures, such as sequential and indexed, are useful at times, their presence in the software support is not absolutely essential so long as a good, efficient scheme for direct access is implemented.

To the extent that the data management scheme chosen implies the need for utility programs, these utilities should be present. Extremely rapid access methods should be provided to avoid "choking" an application on its own real-time output.

Program Management

Basically, this implies a mechanism to load and execute programs (and to get rid of them afterward). However, MSRF will assuredly require multiprogramming, so the problem becomes one of loading and executing many programs in several computers. If the approach selected implies the need for "dynamic storage allocation," then a good implementation of this concept should be present. Furthermore, there must be mechanisms for management of the concurrent execution of many programs, for intercommunication between these programs, and for their sharing of resources with provision for temporary exclusive use. There are literally hundreds of ways in which these features can be implemented, and there exists no absolute frame of reference in which they may be compared. Each application defines, by its requirements, a comparative frame of reference such that some methods will clearly be seen as superior to others. Clearly, no single method will be the best for all applications. We feel that the method of program management used for a given MSRF application should be up to the programmer. If this is infeasible in the system which is selected, one would at least hope to find evidence that considerable thought had been applied to each of these areas and that considerable capability had resulted.

Multiprogramming systems are often intended to support multi-job batch or multi-user time-sharing. For this reason, one of their most compelling design requirements in the area of program management is that they should build conceptual "walls" around each program. In such an environment, programs can be unaware of each other's existence, and, most importantly, each program is supposed to be absolutely protected from damage due to any conceivable error on the part of another program. It is of great importance to remember that this is *not* an appropriate design requirement for MSRF. The safety features are unnecessary because MSRF will be

running singular, checked-out applications, and they are in fact undesirable because of their expense in space and time and because the restrictions they imply create unnecessary restrictions for the programmer. It is far more important that attention be given to intercommunication between programs in separate processors, which includes the ability for programs in one processor to use peripherals attached to another. The system should make special efforts to support the use of reentrant coding.

UTILITY PROGRAMS

Each operating system design implies a set of routine maintenance or administrative chores. Often, these have to do with manipulating or backing up mass storage files. Whatever routine work the operating system might demand of the user should be supported by appropriate, easily used utility programs. Since the interchange of files with other computer systems has been postulated as one of the requirements for MSRF, utilities should exist to aid in this operation.

LIBRARY ROUTINES

It is possible to predict the nature of a set of modular pieces of software which will probably be quite useful to MSRF applications. Hopefully, these would be implemented in a skeletal fashion so that they could be altered to fit the needs of given applications.

Mathematical Routines

Such functions as square root, logs, exponentials, trigonometric functions, and vector arithmetic will probably be required for many MSRF applications. Random number generators and various statistical functions are often used as stimuli for simulations. Sorting and searching algorithms, both for mass storage files and in-core tables, are of great potential utility. Basic signal-processing algorithms, such as filters, may be helpful in dealing with trackballs or joysticks.

Graphics

A comprehensive graphics package should be prepared. It should be capable of exercising all the features of the primary display system and of the plasma panels. Specialized graphics modules should be prepared for producing standard picture elements such as NTDS symbols.

Terminal Modules

Specialized, skeletal software should exist for interfacing to each of the input or output devices which may be configured into a console.

Personality Skeleta

For systems such as NTDS, or certain sonars, it may be worthwhile to formulate open, easily changed software packages which create the basic environment and behavior for simulating them. These will not be trivial efforts, and each should probably be created as the need to deal with each system arises. Their value for subsequent experiments involving the same systems is obvious.

Debugging Aids

Tools which serve to improve programmers' efficiency in program checkout are always welcome, and some reasonable quantity of these should be provided.

HARDWARE DIAGNOSTICS

Routines to test processors and standard peripherals should be obtained from the computer manufacturer. Diagnostics for other parts of the system, such as the interprocessor interfaces and virtually the entire consoles, will need to be written. Such routines have the responsibility for exercising hardware, detecting malfunctions, and analyzing them to aid in repair.

DOCUMENTATION

Several levels of documentation will be required:

Internal System Documentation

This should contain sufficient information to enable those who study it to alter or maintain any part of the system software. Flowcharts are not required, but glossaries of symbolic names, diagrams of data bases, and thorough functional descriptions in prose are of vital importance.

Programmer's Reference Manual

This should describe the *use* of all system software. Each routine or package should be discussed in detail, including limitations, effects of use, peculiar behavior, accuracy, timing considerations, and any other information of interest to a programmer who must employ the tools he is given. Examples of the use of each part of the software should be included.

Introductory Course Material

It is often difficult for a new programmer to acquire a basic understanding of a new system in a short time. We would recommend that MSRF should acquire an introductory course in the capabilities and use of the entire facility, for both programmers and experimenters. This course might employ a 30-60 minute videotape presentation, an introductory text, and a hands-on demonstration program.

VOICE COMMUNICATIONS

Our analysis has indicated that providing the necessary communications for simulating sonar (ASW) and NTDS applications in the MSRF will satisfy all communications requirements for simulating other Navy systems of interest. ASW, NTDS, and the recommended MSRF voice communications systems are described in this section.

ASW Communications

INTERNAL COMMUNICATIONS

There are two primary internal ASW circuits, both of which are implemented using sound-powered phones. The two circuits are the ASW Control Circuit, designated the 1JS, used for exchanging control and decision type information among the ASW Officer, the Sonar Supervisor, the CIC Evaluator, and the conning officer on the bridge; and the CIC (formerly Sonar) Information Circuit, designated the 61JS, in theory a one-way circuit used for sending target information (basically target range and bearing) from Sonar to CIC and the Bridge.

Other internal communications used during ASW operations include a Sonar Announcing System, designated the 29MC, used for making initial contact detection reports and passing target information from Sonar to the Bridge and CIC until the 1JS and 61JS are manned. The 29MC is a PA (loudspeaker) type system. The microphone is controlled by a footswitch located under the bullnose of the sonar console. Best practice dictates that the 29MC not be used after the regular sound-powered phone circuits are manned, but most ships use it to make lost and regain contact reports each time these events occur even when the 1JS/61JS circuits are manned.

The Captain's Command Circuit, designated the 21MC, is used during various evaluations including ASW. It is a two-way intercom system used by the officers in charge of various work stations to talk directly to one another. The system uses a press-to-talk switch and has several push-button type selector switches to permit the operators to select one or more stations.

EXTERNAL COMMUNICATIONS

There are two primary external communications circuits used during ASW, both of which are implemented using UHF voice radiotelephones. The SAU (Search Attack Unit) Tactical Circuit is used for exchanging ship control and important tactical information between the command stations, i.e., bridges and CICs, of ships participating in a joint ASW operation. The SAU CI (Search Attack Unit Combat Information) circuit is used to exchange target information among the CICs of ships participating in an ASW operation.

Several other radio circuits may also be operated during ASW operations, e.g., Screen Common, used for exchanging tactical information among the units screening a main body, and the ASW Air Control Circuit, used for directing the movement and employment of ASW helicopters and fixed-wing aircraft. The total number of radio circuits used during a particular tactical operation is primarily a function of equipment availability.

The key consideration as far as the MSRF is concerned, however, is that a particular individual is seldom required to communicate on more than one radiotelephone circuit and one sound-powered circuit at any given time. He may, however, be able to hear announcements made on the PA-type circuits and on other radiotelephone circuits when they are fed into a work station by loudspeaker. Sonar room personnel will not be involved in these external communications circuits. Figure 18 shows the communications channels used in ASW that are relevant to the MSRF.

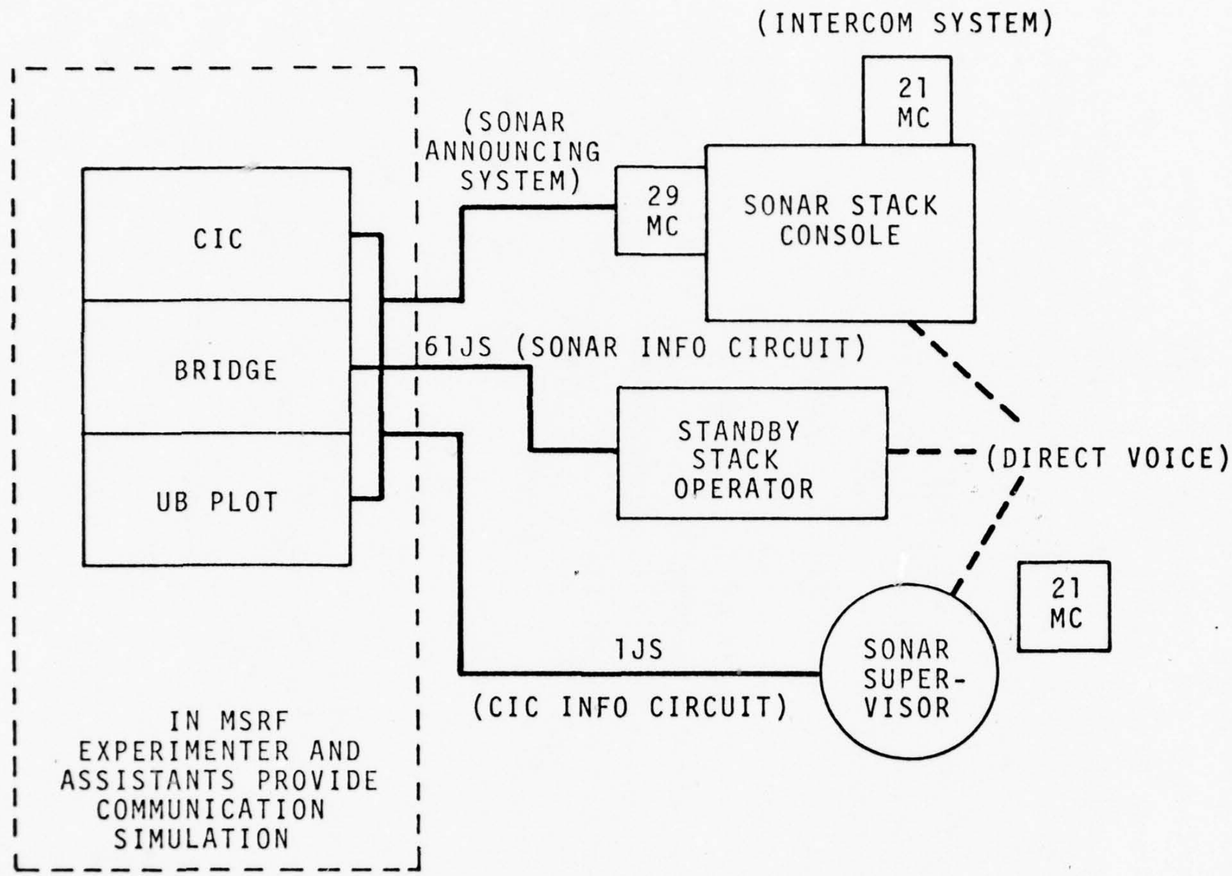


Figure 18. Schematic representation of typical sonar voice communications.

NTDS Communications

INTERNAL COMMUNICATIONS

There are two key differences between NTDS internal communications and the internal communications used with most other shipboard systems. The first is the communications control panel located on the NTDS operator's console. This control panel allows the operator to select the particular station(s) (other console operator[s]) he wishes to talk to. He may, by appropriate selection of buttons on his control panel, communicate with any or all other operators.

The second difference between NTDS and other internal communications systems is the NTDS position indicator. The position indicator is a 1/2" square symbol which can be displayed on the operator's CRT and transmitted to the CRT of the station(s) selected on the communications panel. The location and movement of the position indicator is controlled by the operator's trackball (following actuation of the position indicator push button). This capability can be simulated in MSRF with the recommended display and computer system.

EXTERNAL COMMUNICATIONS

Several voice radio circuits are employed during NTDS operations. As in the case of ASW, a particular individual will seldom be responsible for more than one radio circuit, although he may hear messages received on other circuits and broadcast on loudspeakers in CIC.

The key difference between NTDS external communications and non-NTDS external communications concerns the use of computer controlled links. There are three of these links at present: Link 11, used to exchange track data, engagement status information, and tactical orders among NTDS Participating Units' computers; Link 14, used to transmit tactical data from NTDS to non-NTDS ships in the form of teletype

messages; and Link 4A, used to transmit air control orders from NTDS shipboard air intercept controllers to fighter aircraft. The voice communications channels relevant to the MSRF are shown in Figure 19.

MSRF Communications System

In the MSRF each console will have six sound-powered communications circuits which will allow conversations between two or more console operators simultaneously and also will allow communications with simulated areas outside the experimental room such as the ship's bridge, CIC, other ships, etc. Two circuits are essential because NTDS requires an internal and an external communications channel. The four additional circuits are to allow for experimental flexibility in communications. Additionally, each MSRF console will require an intercom to simulate the 21MC and 29MC circuits. At least two additional communications stations should be located in the MSRF experimental room for additional personnel such as the standby sonar stack operator and the sonar supervisor. Two communications stations should also be located in the experimenter's control station. These stations will allow the experimenter to monitor conversations between subjects and may be used to give instructions. Also, these stations will act as the simulated external communication stations during experiments.

To the subjects in the experimental room, the communications available will be identical in function to those available for actual surface ship systems. Each console will have a headset and a communication select panel. In addition, a footswitch which is used in sonar applications for the 29MC communication channel should be provided at each console. The subject console operators will be able to select one or more stations to communicate with. When areas outside the simulated area are selected they will be routed to the communication

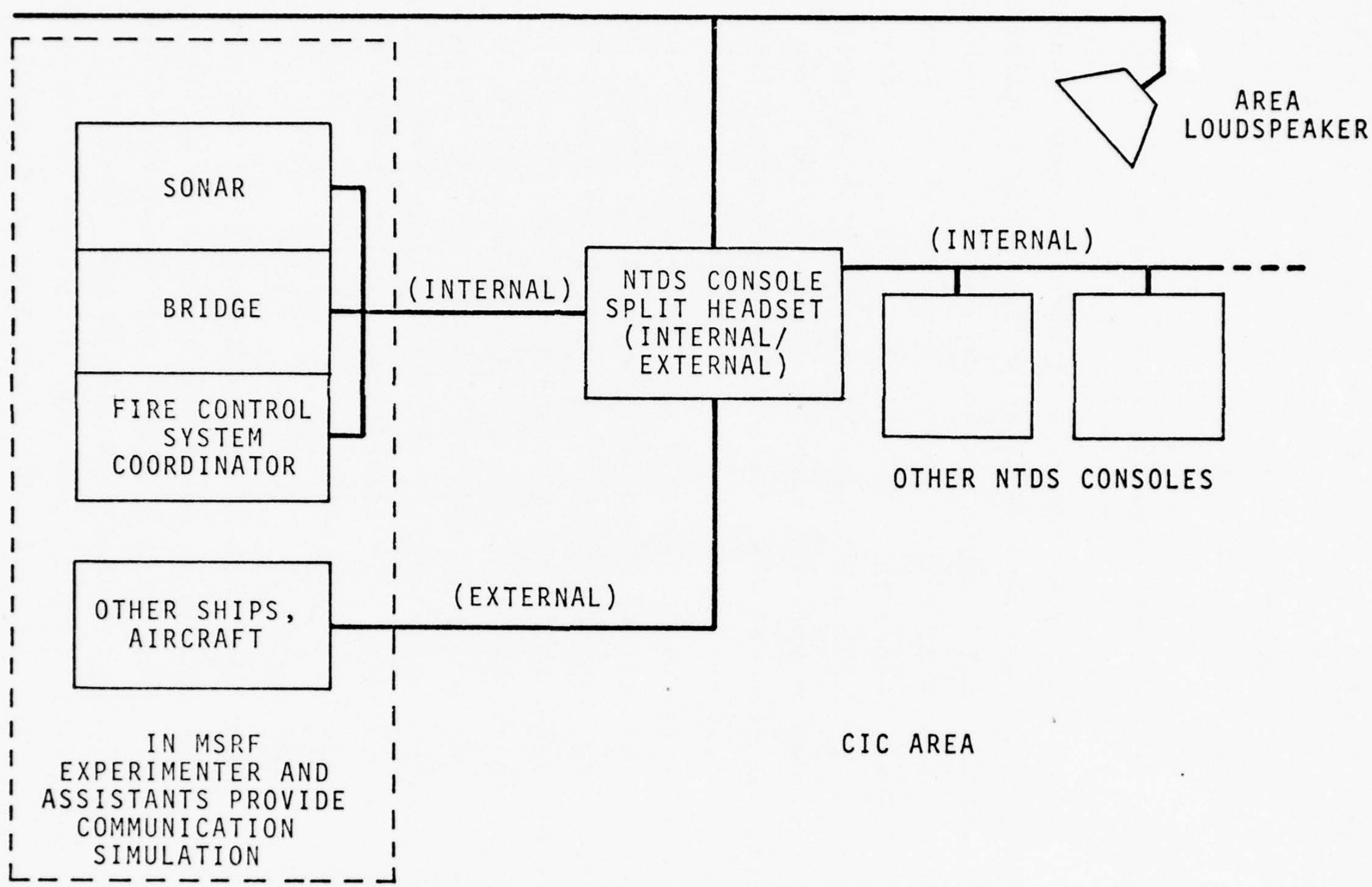


Figure 19. Schematic representation of NTDS console voice communications.

station(s) in the experimenter area regardless of the simulated stations selected through the communications panel.

MSRF Communications Recording

For many experiments, it will be essential to record verbal transactions taking place in the MSRF Communications System to permit post-experiment analysis. Furthermore, in order to establish an unambiguous, permanent temporal relationship between the audio tape voice recordings and the computer data base recordings of concurrent man-machine interface transactions at the subject consoles, it is recommended that a time code generator be employed to supply master time for the MSRF, by means of a time code channel to the voice recorder, a time reference source for the computer system, and various directly viewable time readouts as necessary.

There are a number of alternative methods for fulfilling these requirements. Three technically acceptable alternatives are listed below.

1. Use an 8-track instrumentation recorder and a time code generator. This system would be capable of recording 7 circuits on individual tracks plus the time code.

Cost:	8-track instrumentation recorder (HP 3968 A)	$ 8,520.00
	Time code generator (Moxon 540)	2,600.00
	Total	$11,120.00

2. Use a 4-track instrumentation recorder and a time code generator. This system would be capable of recording three circuits on individual tracks plus the time code. A mixer would be required if all 7 circuits were to be recorded.

Cost:	4-track instrumentation recorder (HP 3964 A)	$5,860.00
	Time code generator (Moxon 540)	2,600.00
	Total	$8,460.00
	Mixer	100.00

3. This is the same as #2 except a 4-track entertainment recorder is used instead of a 4-track instrumentation recorder.

Cost:	4-track entertainment recorder (Sony TC-277-4)	$ 470.00
	Time code generator (Moxon 540)	2,600.00
	Total	$3,070.00
	Mixer	100.00

We feel that the entertainment recorder alternative (#3) would provide acceptable performance at a considerable saving in cost, and we therefore recommend it.

A schematic representation of the recommended MSRF communications and voice recording system is shown in Figure 20. The approximate cost of the basic system components is detailed below.

Sound-powered headsets with boom microphone (David C. Clark Co., Inc., Model 10SB-A)	7 @ $ 140	=	$ 980
Push-button stations	7 @ $ 100	=	700
Intercom stations (Talk-a-Phone, Model K-AC-510)	6 @ $ 100	=	600
Recorder (Sony TC-277-4)	1 @ $ 470	=	470
Time Code Generator (Moxon 540)	1 @ $2,600	=	2,600
Total			$5,350

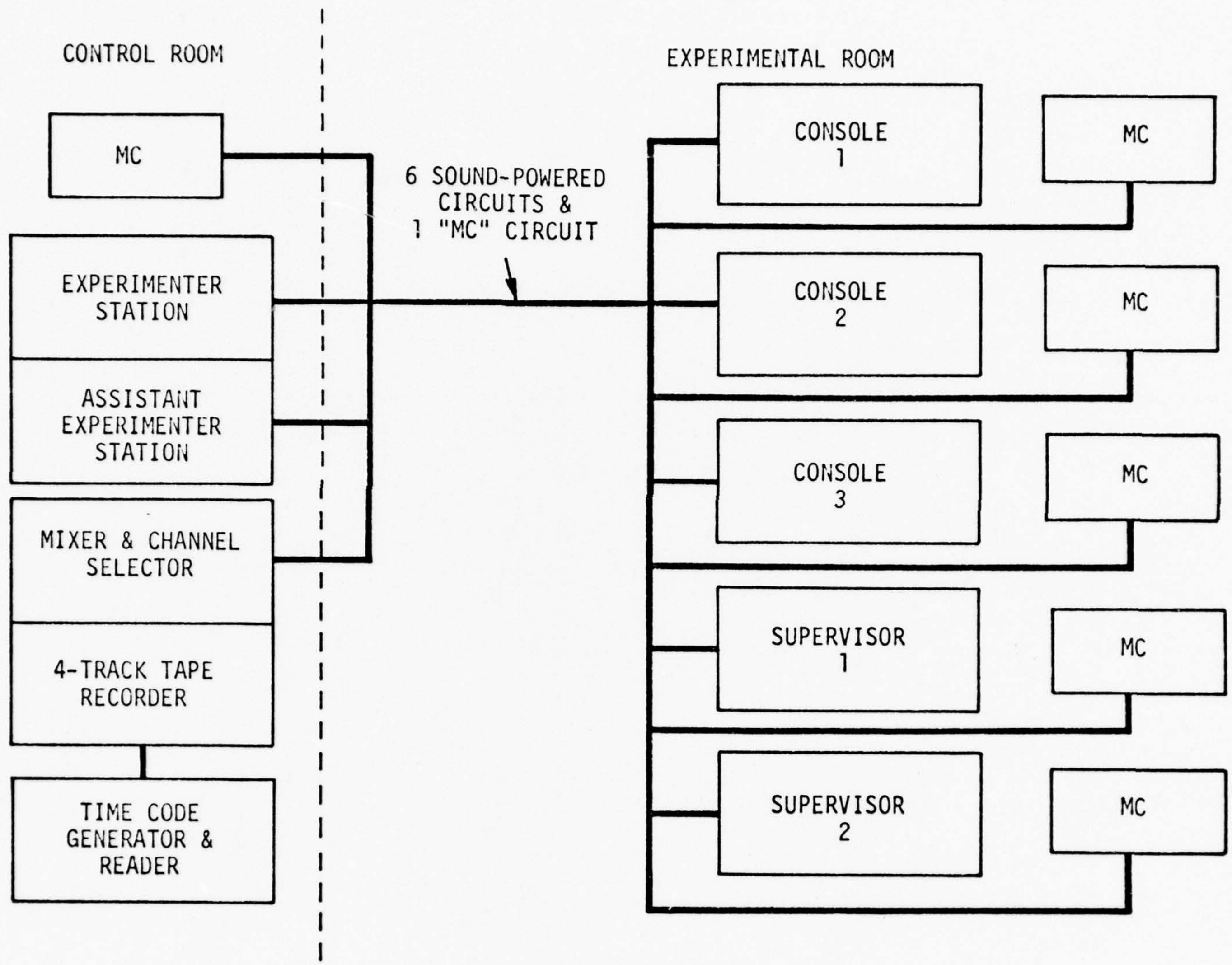


Figure 20. Schematic representation of MSRF communications and voice recording system.

Preceding Page Blank - Not Filmed

PHYSICAL FACILITY

Experimenter Work Station

The experimenter work station will serve as the central observation and control point for experiments conducted in the MSRF. Because of the presence of a CRT terminal, it may also be used for the preparation and testing of applications programs. As Figure 21 illustrates, the experimenter work station has four principal features--a large one-way mirror for observation of activities in the experimental room, a video monitor by which the experimenter can view any of the displays actually shown at the subject work station consoles, two voice communications stations, and a CRT terminal. Ample table space is provided in front of and behind the experimenter for record-keeping, scenario scripts, etc.

The analysis of the requirements for the experimenter work station and the results of the research support requirements survey indicate that the experimenter will have little occasion to intervene in the conduct of the experiment once it has started. Generally, it is only necessary that the experimenter be able to set a few initial conditions, cause a stop or pause in the ongoing experiment, and restart the experiment with new or different parameters. There is no expectation that the experimenter will be required to provide continual orchestration of the events occurring in the experimental room. Once an experiment has started, the experimenter will be primarily concerned with observing the general progress of the experiment, possibly monitoring summary data displayed on the CRT terminal, and acting as an external voice communication station for the simulation in the experimental room. Since it is possible that more than one external voice communication exchange may be occurring at the same time, an additional voice communication station has been provided for an assistant.

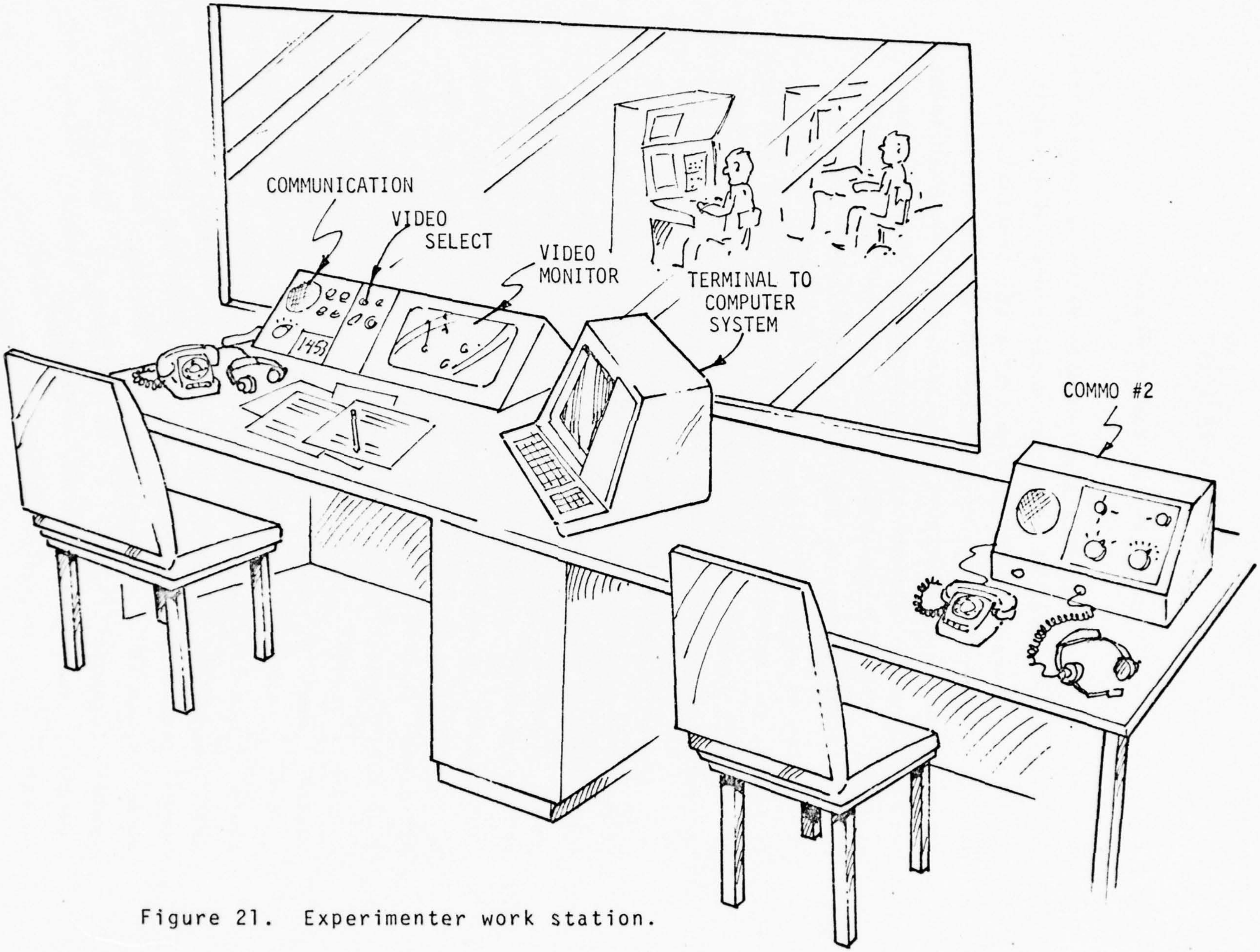


Figure 21. Experimenter work station.

The video monitor will be a repeater for the displays on the work station consoles in the experimental room. A simple control box consisting of a 6-position rotary switch will allow the experimenter to select the display he wishes to have repeated on his monitor. Voice communications will be available through a sound-powered headset, and the voice circu. controls will be identical to those present in the experimental room.

It is recommended that the CRT terminal be functionally equivalent to a VT-52 manufactured by Digital Equipment Corporation, which has a cost of $2,200. This terminal has a 8.6 x 4.5 inch screen and is capable of displaying 24 lines of 80 characters each. Other characteristics were described in the computer system section. An alphanumeric keyboard will allow the experimenter to enter control commands and parameters to the computer system. It is anticipated that most experiments will require some type of statistical information, computed in real time, to be displayed on the CRT to inform the experimenter about the progress of the experiment. The type and form of this information will depend on the requirements of the particular application. A line printer located in the same room can produce more extensive data during the course of the experiment or summary data at the end of the experiment.

Security Requirements

The results of the survey indicate that classified material in either hard copy or electronic form would be used only occasionally. Therefore, there appears to be no reason to plan for either physical security for the storage of classified documents or electronic security, i.e., shielding of equipment to prevent electronic emissions. It is assumed that hard copy classified material would be under direct control of an individual and would be stored, when

not in use in the MSRF, at some other location within NPRDC. When a research project which definitely requires use of classified material is scheduled in the MSRF, it may be desirable at that time to include an approval security container in the MSRF. It is expected that this container would be used only for temporary daily storage of the materials as necessary.

According to Naval Electronics Systems Command, NAVELEXINST 05510.2, 11 March 1969, the MSRF would qualify for exemption from TEMPEST protective measures. In accordance with Chapter 5 of the referenced document, the exemption is based on the anticipated low average daily utilization of the MSRF for classified data processing and the location of the MSRF well within the boundaries of a large controlled access area, i.e., NPRDC.

Physical Facility Development

FLOOR PLAN

The MSRF will be located on the ground floor of a refurbished temporary barracks building, No. 337, located at NPRDC, San Diego, California. The total usable area available on the ground floor of this building exclusive of lavoratory and stairways is 1,644 square feet. The MSRF will share the ground floor with the Automated Training and Testing System (ATTS) computer. The MSRF and ATTS facilities will be completely independent. ATTS will require 478 square feet. The MSRF will utilize the remaining 1,166 square feet. Figure 22 is a cutaway perspective view and Figure 23 is an architectural plan view of the ground floor of Building 337. The space utilization for the MSRF and ATTS is shown in these figures. The MSRF area will be divided into three rooms. The first will house the computer mainframe and mass storage device as well as storage cabinets. This room will be 15.5 x 14 feet for a total area of 217 square feet. The second room

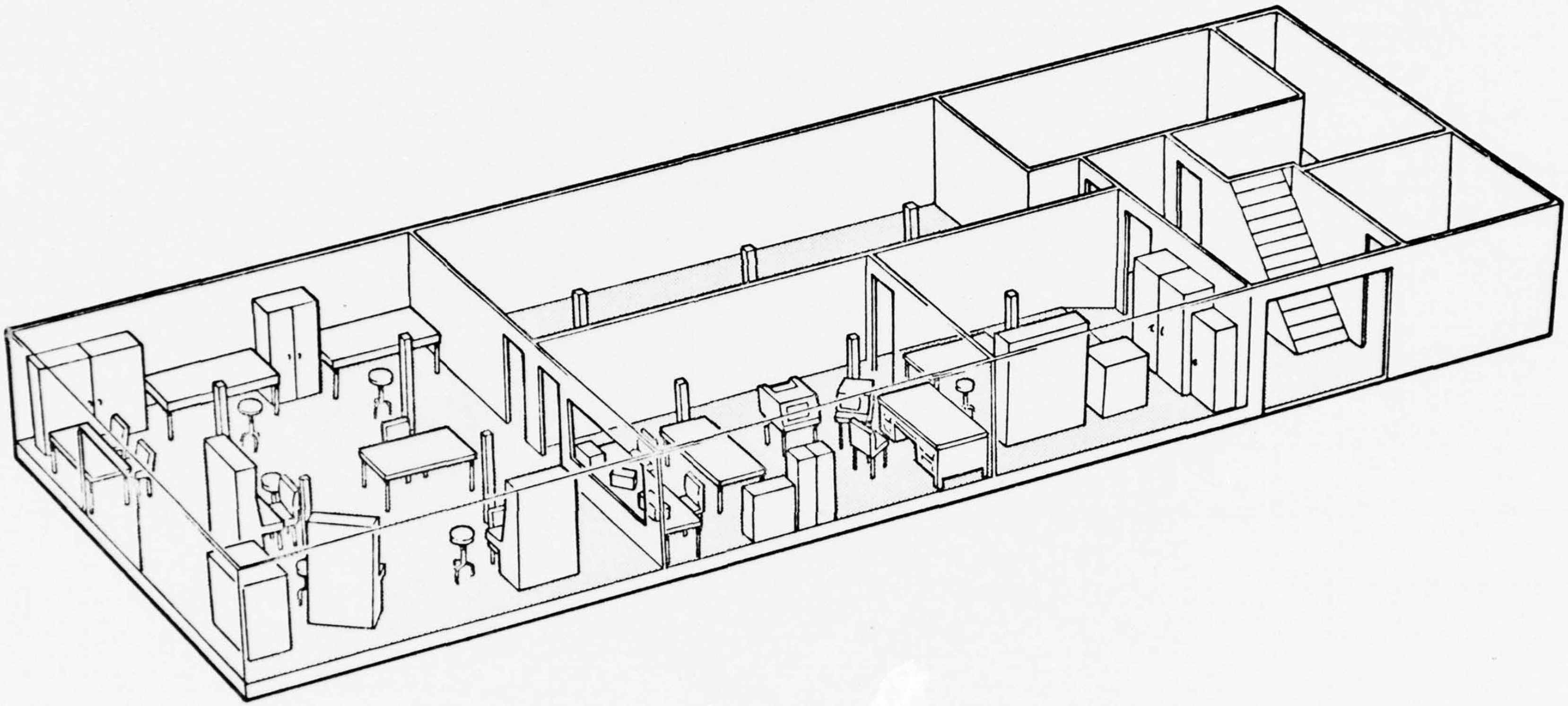
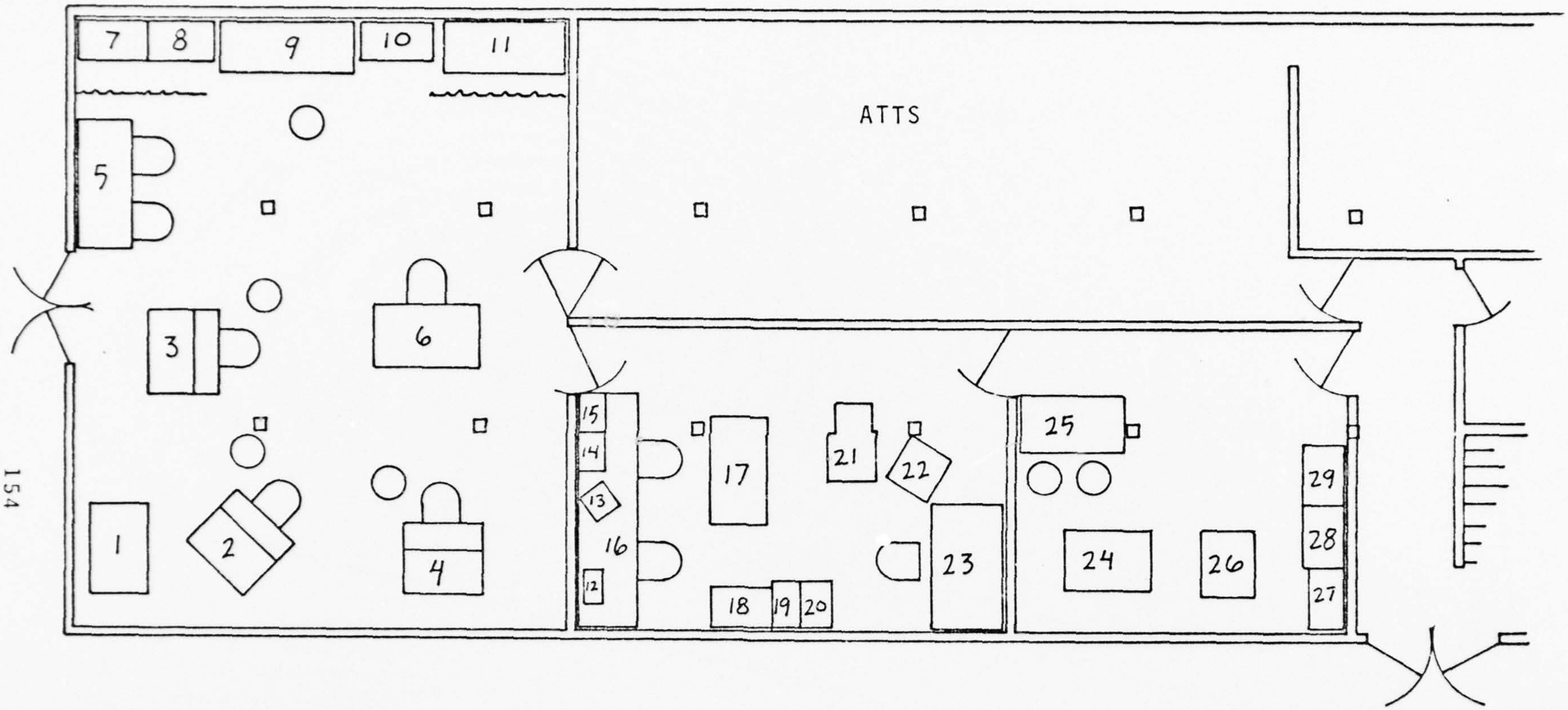


Figure 22. Cutaway perspective view of the MSRF physical facility.



1. DISPLAY PROCESSOR
2. CONSOLE
3. CONSOLE
4. CONSOLE
5. TABLE
6. TABLE
7. STORAGE
8. STORAGE
9. WORK BENCH
10. STORAGE
11. WORK BENCH
12. COMMO
13. CRT
14. VIDEO
15. COMMO
16. TABLE
17. TABLE
18. STORAGE CABINET
19. FILE CABINET
20. FILE CABINET
21. PRINTER
22. CRT TERM
23. DESK
24. COMPUTER
25. TABLE
26. DISK DRIVE
27. TAPE STORAGE
28. STORAGE CABINET
29. STORAGE CABINET

Figure 23. Plan view of the MSRF physical facility.

is the experimental control and program preparation room. It will be 20 x 14 feet for a total area of 280 square feet. This room will contain the experimenter work station, program development station, and the system line printer. In addition, desks, tables, and storage cabinets will be present. The third room is the main experimental area and will be 28.5 x 23.5 feet for a total area of 670 square feet. This room will contain the subject work station consoles and a small shop area at one end of the room. When the room is being used for the conduct of experiments, the shop area will be closed off by accordian wall partitions.

BUILDING REFURBISHMENT

The following is a description of the requirements to prepare the building for the MSRF facility. The requirements include general refurbishment of the building and installation of raised computer flooring; provision of adequate power, air-conditioning, and lighting; and the required furnishings for the building. The general refurbishment plan and installation of computer flooring include both the MSRF and ATTS areas. The additional requirements refer to the needs of the MSRF only.

The details of the refurbishment of the building will be developed by the contractor responsible for the acquisition and integration of the MSRF equipments and the Navy Public Works Office. In general, the refurbishment plan will call for the removal of all windows, insulation of the exterior walls and floor space, and the installation of interior walls as shown in Figure 23. In addition, raised computer flooring will be added at this time. Computer flooring is available from Al-Lee Construction Co., Inc., 4609 W. Jefferson, Los Angeles, California. If the flooring is installed by an outside contractor, the cost will be approximately $7 per square foot for a total of approximately $11,000. The computer flooring is required so that all cabling between the computer, its

peripherals, and the subject work station consoles can be physically protected and allow for rapid reconfiguring and positioning of the computer equipment and the work station consoles. Installation and maintenance will also be greatly facilitated by the use of this flooring.

POWER REQUIREMENTS

The MSRF requires an electrical power system with a minimum capacity of 60 amps distributed by a 115/208 volt, 3-phase, 4-wire system. This is a standard type of power system. The exact number of 115 volt and 208 volt outlets and their positions will be specified by the contractor responsible for the physical development of the MSRF. Power requirements for ATTS and the air-conditioning system are not included.

AIR-CONDITIONING

The air-conditioning capacity for the MSRF will be approximately 55,000 BTU per hour. This estimate is based on the published heat load figures for the Digital Equipment Corporation equipment and an allowance of 660 BTU per hour for each person occupying the MSRF. The cooling capacity for the occupants is based on a maximum of 15 people in the MSRF. It is *estimated* that the ATTS area will require approximately 40,000 BTU per hour cooling capacity. The air-conditioning system compressor and condenser will be located adjacent to the building.

LIGHTING

Standard office fluorescent lighting will be required throughout the MSRF, including the experimental room. The fluorescent fixtures in the experimental room should each have an individual switch. In addition, an incandescent system must be provided in the experimental room. The incandescent lighting must be controlled by an adjustable autotransformer. It is recommended that an incandescent fixture be mounted

approximately every 10 square feet within the experimental room. The florescent system will allow general office lighting for maintenance and many experimental situations and the incandescent system will provide a means of conducting experiments under reduced lighting conditions.

FURNISHINGS

The furnishings recommended for the MSRF consist of those items shown in the floor plan in Figure 23. The majority of the furniture will consist of standard Government office items including:

6 metal general-purpose storage cabinets, 36" wide x 24" deep x 72" high

3 small tables, 36" long x 60" wide

1 large table, 36" long x 84" wide

1 double-pedestal desk, 60" long x 34" wide

2 lockable 4-drawer file cabinets, 18-1/4" wide x 28-5/8" deep

1 computer tape storage rack, 36" wide x 19" deep x 84" high

2 electrical workbenches, 34" wide x 72" long

6 swivel chairs, padded with metal arms

6 chairs, padded with metal arms

2 chairs, padded, armless

1 extra-large table, 36" wide x 132" long (This is the experimenter's work station table; it may be necessary to have it custom-built to accommodate the video monitor, communications stations, and computer control terminals. Two standard small tables may also suffice for this application, or it could possibly be constructed from the same modular components as those used in configuring the subjects' work stations.)

It is expected that many of the chairs and tables can be used for multiple purposes. For example, the tables and chairs located in the experimental room may be used between experiments by the MSRF staff. Additional miscellaneous furnishings such as coat racks and additional small tables or chairs can be added after the facility is manned. This will depend upon the space available and additional needs that arise at the time of initial use of the MSRF.

Many of the details of the refurbishment, location of power outlets, air-conditioning duct locations, and positioning of the lighting fixtures have not been specified in detail. The exact specifications for these items will be the responsibility of the contractor who, in conjunction with the Public Works Office, develops the refurbishment plan. It is anticipated that approximately 9 months, minimum, will be required to plan and execute the refurbishment.

STAFFING

The general functions of the MSRF staff include management of the facility, scheduling and coordination of its use, the development and documentation of systems and applications programs, software and hardware configuration control, and maintenance and reconfiguration of the computer and console equipments.

It is generally expected that users will provide guidance to the MSRF staff on their research plans and the requirements for support. In some cases, the MSRF Programmers will have complete responsibility for the development of a particular application program. In other instances, the user may provide in-house or contractor personnel to assist in the development of the software. It is probably not reasonable to expect the MSRF Programmers to have the time to do all software for all experimental applications. Also, while the MSRF staff will have the full capability of reconfiguration of the computer system and the work station consoles, they cannot be expected to create sophisticated or very special-purpose equipments for a particular experiment.

Based on these expectations, it is recommended that the MSRF staff consist of four permanently assigned personnel: (1) the MSRF Facility Director, (2) Senior Systems Programmer/ Deputy Director, (3) Systems and Applications Programmer, and (4) Electronics Technician.

The MSRF Facility Director will have overall responsibiity for the management of the facility, supervision of personnel, and scheduling of developmental and experimental work within the MSRF. The Facility Director will also provide direct liaison with NPRDC management and the division heads within NPRDC. The Facility Director should have a good understanding of the purpose and methods of experimental work that will be carried out in the MSRF and professional competence in computer

science. The Facility Director should be, of course, thoroughly familiar with the computer equipment and software employed in the MSRF.

The Senior Systems Programmer/Deputy Director will have primary responsibility for software development and documentation, software and hardware configuration control, and direct supervisory responsibility for the day-to-day activities carried out within the MSRF. This individual should have professional competence in computer science and will be the chief architect and programmer for MSRF systems software. He will be the chief assistant to the MSRF Facility Director for the planning of software and equipment development and be responsible for the estimation of time and level of effort required for various systems and applications developments. The Senior Systems Programmer/Deputy Director also will have primary responsibility for the supervision and approval of software developments that will be used in the MSRF regardless of whether the software is developed in the MSRF, outside the MSRF but within NPRDC, or outside NPRDC.

The Systems and Applications Programmer will have primary responsibility for the writing and testing of systems and applications programs as directed by the Senior Systems Programmer/Deputy Director. He, as well as the Senior Systems Programmer/Deputy Director, should have considerable experience in the application of computers to real-time problems, and should develop thorough working knowledge of those Naval systems of primary interest to NPRDC.

The Electronics Technician will be responsible for all physical wiring of the electronic devices in the work station consoles and the interfacing of these devices to the console computer, the main system computer, and the display processor. Essentially, his responsibility will extend to all wiring and equipment which is not integral to the computer systems or the display processors themselves. The Electronics

Technician should have seasoned experience in the design and wiring of digital circuitry and a working knowledge of analog and electro-mechanical circuitry. Under the supervision of the Senior Systems Programmer/Deputy Director, the Electronics Technician will be responsible for documenting all existing, new, and changed wiring and circuits. An additional duty of the Electronics Technician will be to reconfigure the work station consoles as required for current experiments.

When requested, the Electronics Technician should be able to provide information about the required electronic characteristics of new devices which may be incorporated in the work station consoles or interfaces. The Electronics Technician need not necessarily be a full-time permanent member of the MSRF staff. It would be entirely possible to use an individual who has a permanent job assignment outside of the MSRF but would be on-call as needed. In this case, however, it would be important that the same individual be used for all changes to the MSRF electronic wiring and circuits for purposes of efficiency and configuration control. It would not be desirable to allow any one of several electronics technicians to work on the MSRF equipment. The best possible arrangement would be to have the Electronics Technician permently assigned to the MSRF and be assigned additional duties outside of the MSRF rather than the reverse.

Since the Electronics Technician's responsibilities will necessarily include troubleshooting of a complex system and some design work, it should be recognized that a fairly high-level developmental technician or near-engineer will be required. This requirement could be relaxed somewhat if the Technician could draw on the expertise of engineers on the NPRDC staff; however, it is our experience that it is less expensive to acquire a good technician than to attempt documenting the system so thoroughly that the need for autonomous problem-solving capability is alleviated.

PRECEDING PAGE BLANK - NOT FILMED

ACQUISITION AND DEVELOPMENT PLAN

Preliminary Discussion

We have been charged with recommending a course of action which satisfies three contraints: (1) to spread development costs equitably over a 3-year period, (2) to enable the use of MSRF for some nontrivial research project at the end of the first of these years, and (3) to minimize costs.

During the course of this study, technical and practical problems have continually arisen to diminish the feasibility of meeting all these constraints as well as we may have desired at the beginning. The central, recurring problem has been that these constraints have increasingly proven to be mutually exclusive. The worst problem area has been achievement of a working facility within 1 year without causing either increased overall costs or unequal distribution of costs over time.

The nature of this problem is best discussed in terms of the unscaled PERT chart in Figure 24. The times for several key tasks in this chart are rather long, and those which are capable of acceleration at all can only be shortened at great expense. Item A, primary display system procurement initiation, is bound to take at least 1 month. We cannot image a shorter interval here; final discussions and agreement with Hazeltine on a final configuration and its price will take a minimum of 2 weeks, and we would be astonished if they could be given an order in less than 1 month. Indeed, Item A could take more than a month.

Item B represents the time delay between order and delivery of the display system. Under ideal conditions, Hazeltine has given us to understand that this would be 6 months. By ideal conditions, we mean placing an order at the time they are beginning a production run. Such a time, it is claimed, would be October-November 1977. This time could be considerably longer were an order placed under non-ideal conditions.

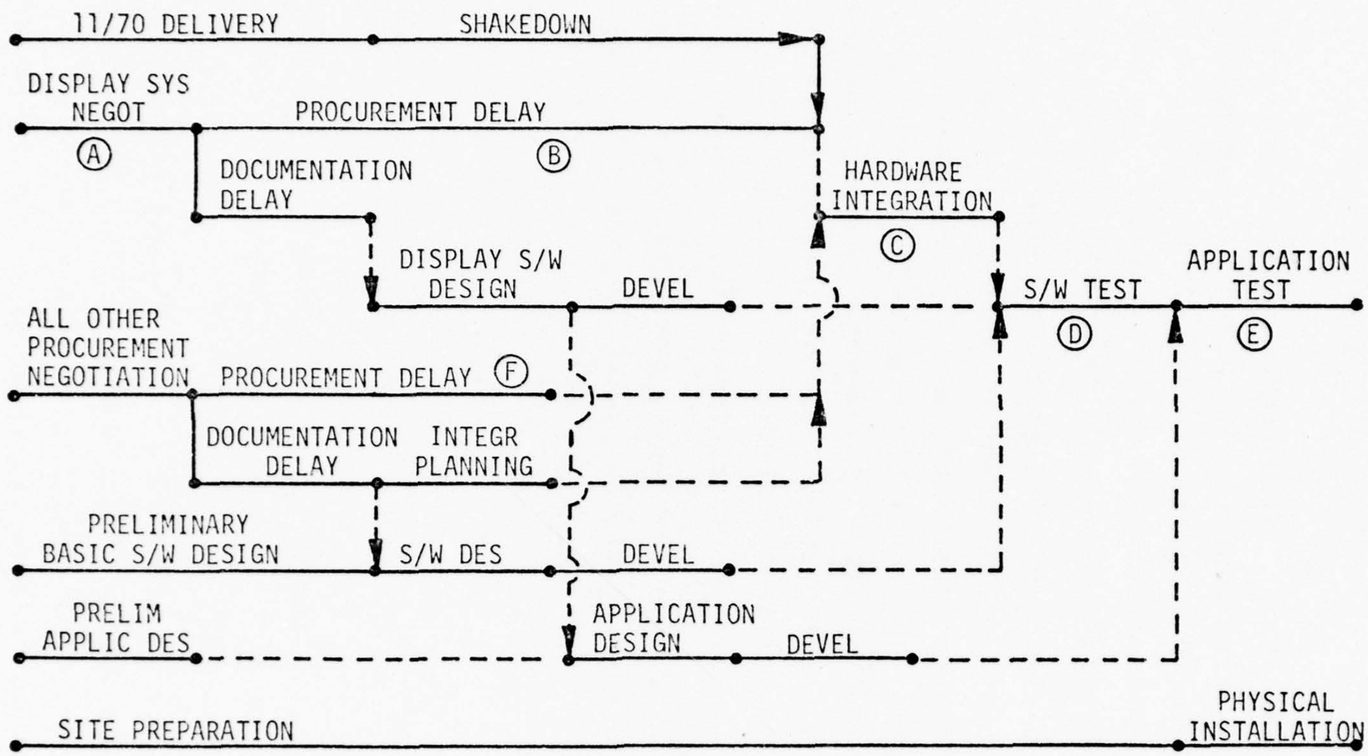


Figure 24. PERT chart.

The delay for procurement of other required hardware components, Item F, varies from 60 days to 6 months. Even when dealing with DEC, we have witnessed additional delays related to shipping foul-ups, errors made by salesmen, and other unpredictables.

Thus, even with ideal conditions and careful and agressive management of the procurement effort for off-the-shelf equipment, it approaches impossibility to collect all of these components in one place within less than 7 months. Further, this period could easily be extended by uncontrollable factors.

Proceeding toward the performance of an experiment, the critical path next encounters the problem area of hardware integration (Item C). Construction of initial consoles and assembly of the console computers can proceed on a fairly independent basis as the components arrive, as can design and construction of the generalized interface. Testing of the consoles will require some other computer to provide mass storage; this does not necessarily need to be the 11/45. However, the console(s) cannot be considered complete until the primary display system is delivered, integrated, and tested. Furthermore, the hardware integration task will certainly remain incomplete until the interprocessor communication link has been developed and tested, and the additional peripherals for the 11/45 have been acquired, installed, and tested. These tasks cannot be performed until the 11/45 is available without reservation, meaning that the 11/70 for ATTS has been installed and shaken down. It is our understanding that the ATTS computer will be delivered some time in January or February of 1978. If the MSRF implementation effort were initiated in October 1977, the release of the 11/45 by ATTS in February or March would be soon enough to avoid further extension of the critical path. Assuming this schedule, we feel that only with extreme good fortune could all of the hardware be considered to be assembled and working within 9 months.

At this point, software testing (Item D) may begin in earnest. Admittedly, some testing can be accomplished prior to hardware readiness; however, many major components, such as the interprocessor communications discipline and the graphics packages, may only be tested after hardware assembly is virtually complete. We feel that at least 2 months should be allowed for this, although the effort could possibly be shortened (at some indirect expense) if the initial demonstration were sufficiently critical.

Thus at the end of Item D, we are already 11-12 months into the project, having at most 1 month for final implementation and testing of the application program for the initial experiment (Item E). If the experiment were relatively simple, this could be accomplished, although very little time would be available to set up any sort of a stable routine for the running of experiments.

Thus, it appears barely possible to conduct an experiment within 12 months if everything goes according to plan. However, three important areas have been omitted from the preceding discussion.

1. It is assumed in the PERT chart that most basic software design and development can be done in advance of hardware delivery. Generally, this is true; however, it is inevitable that hardware, once delivered, will behave differently than described in its documentation. Sometimes this requires coding changes, sometimes design changes. In any event, it is quite likely for additional software development to be required after the hardware has been brought "up." Further, these efforts could result in the requirement that prewritten application code would need to be changed. To minimize the elapsed time until the facility is operational, it is *essential* that these efforts be conducted in advance, yet there always exists the possibility that some parts of the efforts will have to be re-done. Finally, the full development of much of the software support which is ultimately desired cannot be easily or inexpensively done without having the hardware available. It is anticipated that the MSRF software development effort will continue well into the second elapsed year.

2. As indicated earlier, the best estimates we have received suggest that preparation of the building will require a minimum of 9 months. How this affects the critical path will depend on the intentions of ATTS. If the building refurbishment, ATTS' occupancy, and 11/70 installation are ill-timed, the release of the 11/45 could be delayed and might affect the critical path.

3. This entire PERT chart is based on coordinated management and execution of the entire implementation effort. No provision has been made for support of communication paths between multiple contractors, and for rectification of disputes over responsibility or the delays brought about by failures in those communication paths.

Our initial objective had been to submit a plan which would result in a facility with one fully operational console within 1 year. While it seems that this could be accomplished by an all-out effort beginning in October 1977 if the ATTS computer were available and the building prepared within 6 months, all-out efforts often lead to mistakes. Further, when combining the considerations outlined above with the superficial implications of the PERT chart, it appears that one could predict with certainty that the initial phase of the project would *not* be complete within 12 months.

We therefore recommend that two adjustments be made in the definition of the goals for the first fiscal year of acquisition. The first is to consider stretching the preparation of the building over 2 years. The first year's preparations would provide for occupancy by ATTS and for the construction of the MSRF computer room. This would reduce the elapsed time to ATTS' occupancy and would avoid a situation where unavailability of the 11/45 might interfere with the MSRF critical path. During the second year, the remainder of the MSRF portion of the building would be developed. This would not materially interfere with MSRF progress, since the large experimental area is really intended for team research,

a capability which would not exist until the end of the second year anyway. ATTS could occupy the experimental area during the remainder of the first year on an interim basis if this were desirable.

The second adjustment is to concede that the MSRF software will not be fully operational until the end of the second year. This would not preclude the conduct of the demonstration experiment, but would allow sufficient time for a good job to be done.

Another difficulty occurs in meeting the original constraint of even distribution of expenditure over the 3 years. One major reason for the economy of the primary display system results from the fact that it is, indeed, a single system. The least expensive way in which to acquire this system is all at once, rather than one console at a time. This transfers a large expenditure to the first year which is difficult to compensate for. However, the recommended extension of building refurbishment over 2 years will be of some help. Additional items of procurement might be moved to later years as well, with some sacrifices in both capability and price.

The recommended plan for acquisition and development of the MSRF which follows incorporates these recommended changes in time phasing for building preparation and software development, and represents our best effort to satisfy the original constraints in the context of the findings of our study.

The Plan

REQUIRED MANAGEMENT STRUCTURE

The MSRF, taken as a whole, most definitely constitutes a complex system. As such, its construction should employ the same principles of system management which have proven successful in the solution of similar problems. The implementation effort involves a wide spectrum of interrelated activities, including:

- Procurement of components
- Design and development of components
- Electronic fabrication
- Mechanical fabrication
- Basic software design and development
- Application software design and development
- Hardware diagnostic programming
- Possible experimental design
- Module and system testing
- Documentation
- Training

The necessary key to success in any such diversified effort lies in its management. It has been our experience and observation that such cases emphatically call for unified, coordinated management and execution of the entire implementation effort. Unless one has extreme luck, this requirement is almost never satisfied by simply subdividing the work and letting a separate contract for each work area. History is replete with examples of the failure of efforts conducted in this fashion; perhaps the best publicized example is the fiasco of the Los Angeles County Fire Department Dispatching Center, whose completion was delayed for months while the principal hardware and software contractors pointed the finger of blame at one another for the system's failure to function. Such cases abound because it is difficult, if not impossible, to write separate contracts which *require* cooperation between the contractors and which fix responsibility when the cooperation does not work out. The principal problems arise from two truisms. First, *development* implies that many concepts are in flux, and all parties participating in the development effort must be continually aware of the effects which these sometimes subtle changes have on their own efforts. The cost of maintaining such intimate communication between contractors is high, and each contractor may be expected to do as little as it can get away with in this area. If incompatabilities

appear in their products, who is to say why they developed and, worse, which can be persuaded to change *his* product without protracted argument?

Second, these inevitable fluctuations which occur during a design/development effort usually imply subtle (and sometimes not so subtle) shifts in workload between task areas, and hence between contractors. These shifts again invite dispute. Often in such cases, a single contractor which is merely trying to do a good, professional job and to uphold its contractual obligations is forced, by the inflexible attitudes of the other contractors involved, to perform more than its fair share of the work in order to protect itself from damage to its reputation.

Of course, it is not fair to place contractors in such positions and project organizations such as this seldom result in satisfaction to the customer. The missing ingredient is unified management and responsibility.

It is therefore our recommendation that this plan be executed by, and the responsibility of, a single prime contractor. This contractor would exercise overall design control of MSRF on behalf of NPRDC and would be responsible for resolving any incompatibilities or problems that arose during the effort. It is conceivable that this function could be performed by a detachment within NPRDC; we leave consideration of such an alternative to the Center, as we lack knowledge of the ramifications of such a decision. However, it is our emphatic opinion that final responsibility for satisfaction of the customer's ultimate objectives in building the MSRF should fall on identifiable shoulders, which are hopefully attached to a body capable of satisfying that responsibility.

Another, perhaps novel, recommendation we would like to make is that an employee of NPRDC, hopefully the person who will become the Deputy Director, be assigned during part of the first year's effort to work at the contractor's site.

This would not be necessary during the initial phases of the first year when adequate liaison could be maintained through monthly meetings at NPRDC; however, this person's presence in the later phases would serve several purposes:

1. NPRDC would be aware of the contractor's progress without necessitating extensive reportage.

2. Closer liaison would be possible, resulting in much more rapid resolution of minor implementation issues.

3. The potential training benefits are enormous. The Deputy Director would, upon installation of the system, have a much more thorough understanding of NPRDC hardware and software than could be imparted through briefings or courses, would have hands-on experience with the system, and would understand the motivations for all implementation decisions and design changes.

4. NPRDC would, as a result, have an in-house person competent to train others in use of the MSRF at the earliest possible time.

If full-time assignment of this person were infeasible, monthly (or more frequent) assignments for periods of at least 1 week would still accomplish much.

PHASING

It is expected that the implementation of MSRF can be accomplished within 3 years. The effort can be divided into three phases, each 1 year long, and each having a definable objective. An additional phase, which has been completed, consisted of this design study and of procurement of the PDP-11/70.

The first phase encompasses the majority of the hardware acquisition for MSRF. At its end, the following situation will obtain: The PDP-11/45, with one complete console, will be installed at NPRDC is a partially complete building. The

basic software will be complete, in a preliminary form, and it will be possible to demonstrate a simple single-operator experiment suggestive of later capabilities. Further, it will be possible to demonstrate that the primary display system satisfies all of the requirements stated in this report.

The second phase develops the capability to conduct team research. The second and third consoles are constructed and integrated into the system, and the building is completed. The experimenter station and experimental chambers are fitted out and the shop facility is installed. Software development continues, basic operating software is brought to its final form, and the library of graphic and other software tools is developed. A second, multi-operator experiment is prepared and may be conducted at the end of the phase.

The third phase essentially puts the finishing touches on the facility. Final hardware components are procured and any software found to be lacking in the first 2 years is implemented. Optionally, an experiment involving exhaustive simulation of a major Naval system is prepared.

The MSRF may be considered fully operational only at the end of the second phase, although it could probably be used for minor research efforts during that phase so long as these did not interfere with the software development effort.

DOCUMENTATION AND REPORTING REQUIREMENTS

The general requirements for software documentation were discussed in an earlier section. We strongly recommend that the production of software documentation should occur concurrently with the software development effort. This generally results in better, and more timely, documentation than does a separate effort conducted later. Furthermore, at any stage of development, that stage is fully documented. This is easily accomplished by employing computer-assisted word processing techniques.

Hardware documentation for MSRF should, we feel, be done in a modular fashion. Excellent descriptions of off-the-shelf modules are usually supplied by their vendors and should remain intact. As custom modules are developed, these should likewise be documented in a modular fashion. All such documentation modules should be organized into a well-indexed reference library.

A brochure, describing the overall capabilities of the MSRF, should be produced at the end of the first phase, and updated at the end of the second to become a part of the training materials described at the end of the software section.

During the first phase, the prime contractor should submit monthly progress reports; in addition, monthly meetings should be scheduled at NPRDC since the volume of decisions (and possibly changes) to be made is potentially high. This is particularly true of the first 2 months.

During the second and later phases, monthly progress reporting should continue, with meetings scheduled at less frequent intervals. It seems to us that no extensive final report should be necessary for any of these phases; the effort would be better spent in producing the documentation, brochures, and training materials.

Detailed Implementation

FIRST YEAR

The major problem to be solved in the first year is one of close timing. The year begins with a number of lengthy processes which must be watched closely, since they must reach fruition before much of the development work can be done. This problem is revealed graphically in Figure 25; we see that, after a 2-month flurry of review and procurement activity, a number of delivery delays begin, some of which extend

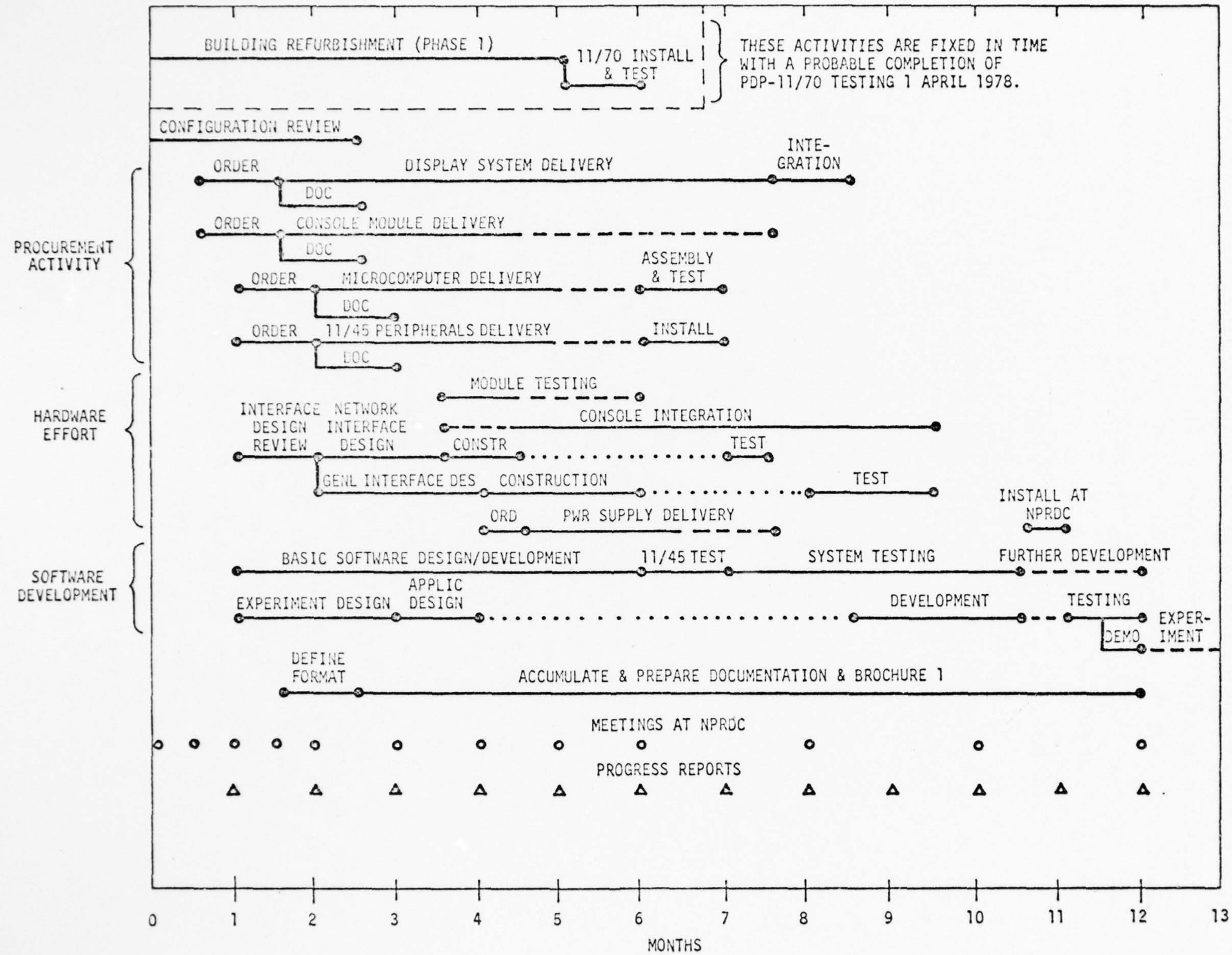


Figure 25. MSRF implementation, Phase 1.

into the 8th month. While some hardware and software design effort can take place during these delivery delays, the majority of the implementation effort cannot begin until the relevant equipment has been delivered.

After each item of equipment is received, five serial activities must take place. The module must be subjected to independent test and inspection, after which it must be integrated with the other modules that have already arrived. These must be subjected, in concert, to system testing, followed by testing of the relevant basic software. Finally, after the system reaches the necessary state of completion, the application program for the demonstration experiment may be tested.

Underlying all of these efforts are continuing design and progress review and the production of documentation. The progress of this plan will next be discussed on a month-by-month basis.

Month 1

It is critical that the primary display system and other physical components with long lead times be ordered as rapidly as possible. To this end, the first month is devoted entirely to a brief technology review followed by procurement negotiations for the primary display system and for console modules other than the microcomputer. Visits should probably be paid to the major hardware vendors to speed negotiations and to collect information which may influence the final computer configuration. Little technical effort is contemplated during this month since direct participation and some travel should be required of the prime technical movers of the implementation.

Month 2

With the critical procurements hopefully in their final stages, this month sees a final analysis of the computer

configuration and ordering of the console microcomputer and of the peripheral devices to be added to the 11/45. Major procurement activity is concluded, leaving only periodic prodding of vendors and maintenance of an up-to-date schedule of estimated delivery times.

Loading of primary technical participants by procurement matters would diminish during this month, enabling them to embark on preliminary efforts in software and interface design. Work should begin on designing the demonstration experiment; this should be done early, since the requirements of this experiment will determine what subset of the MSRF software will be developed during the first year.

Finally, the format of documentation to be produced should be considered. This should be determined early since a modular documentation approach is contemplated.

Month 3

The central activity in this month is hardware design. Approximately 2 months of design time is anticipated for the non-purchasable modules, namely the interprocessor link and the generalized interfacing structure. This effort should begin early, since it will probably necessitate some minor procurements, although its progress may be impeded by delays in receiving hardware documentation from vendors.

These same documentation delays will restrict non-speculative software development to those areas which do not involve peripheral device characteristics. As documentation is received, it should be incorporated into the library.

Month 4

During this month, the hardware design effort should be concluded. Console modules may begin arriving this early, enabling module testing and console integration to begin.

Presumably, most of the hardware documentation will have arrived by this time so that the software development

effort may proceed with few impediments other than the inability to test anything on the hardware. Intensive software effort will likely be necessary through the remainder of the year.

Hopefully, the design of the demonstration experiment will be complete at this time so that the design of application software can be performed. This activity supplies important input to the basic software development effort.

Months 5 and 6

Hardware construction, module testing, and console integration continue. By the end of this period, the majority of console components other than the microcomputer and primary display system should have been received.

Preliminary software development, but no testing, continues.

Month 7

Our experience with DEC causes us to expect that the console microcomputer components should have been received by now. The hardware effort centers around assembly of the microcomputer, testing, and integration into the console.

If all has gone well with ATTS, the PDP-11/45 should become available, allowing software testing to begin on the supervisory computer. Concurrently, the 11/45 peripherals, most of which should have arrived, may be installed and tested.

Month 8

This month should mark the end of major procurement activities. If it doesn't, it seems that the schedule will inevitably slip. Integration and testing of the primary display system begins, and the intercomputer network interface is tested. Software testing reaches the multicomputer stage.

Month 9

Installation of the primary display system is completed and the testing of graphics software begins. The software environment should be well enough defined that implementataion of the application program for the demonstration experiment may begin. Overall system testing continues.

Month 10

This month sees the closure of all hardware efforts other than corrective measures which may be necessitated by testing. The month is principally devoted to concurrent, synergic development of the basic software and of the application.

Month 11

The first half of this month concludes system and application development which has presumably been done at the contractor's site. During the latter half of the month, the equipment is moved to NPRDC and installed in the partially completed building.

Month 12

Final testing and demonstration of the first experiment's application program is conducted at NPRDC, concurrently with last-minute changes to the basic software and possibly the resolution of any hardware problems which may be detected at this time.

System documentation, including the preliminary brochure but excluding training materials, is completed for delivery to NPRDC during Month 13.

Summary

It will be noted that while Phase 1 of the proposed implementation effort is humanly possible, it contains very little room for error. Some parts of this schedule could be made less critical if the contractor were in a position to test some of

the software and interfaces in advance of hardware delivery (by means of simulating the behavior of the missing hardware), or if that hardware delivery could be accelerated. However, the singular nature of the Hazeltine system makes it infeasible (in our opinion) to consider the display software complete until it has been tested on the actual display system. Furthermore, since this device does have certain behavioral characteristics which are difficult to simulate, some software design may be expected after delivery. For these reasons, we doubt that the critical nature of Phase 1 timing can be significantly relieved.

SECOND YEAR

The tasks to be undertaken in Phase 2 are generally simpler and less critical than are those in Phase 1. The central problem lies in converting the single-console demonstration system into a team research facility.

ATTS must vacate the experimental chamber, after which an estimated 10 elapsed months will be spent in finishing the building, fabricating and installing the experimenter console and voice communications system and procuring furniture.

Two more consoles must be fabricated; this should take no more than seven months of elapsed time since the complete display system will have been procured during Phase 1. Interconnecting the new consoles with the supervisory computer should be extremely rapid since no new hardware is involved.

The basic software effort would involve, during the first half of the year, the development of packages, library elements, and utility programs which were deliberately excluded from the first year's objectives. The second half of the year would entail exploration of the multiconsole environment which had been created, testing the viability of the software support and making any necessary changes or improvements.

Probably the most inventive task to be undertaken in the second year will be the design and implementation of an experiment intended to measure team performance. We expect that this could easily take all year, with the majority of the effort spent on experimental design. To permit this time to be spent, the requirements for application programming should be kept fairly simple.

Considering the amount of software development and exploration to be done, together with the chaos in the building for a considerable part of the year, it may be advantageous to move the equipment back to a contractor's site after the first demonstration experiment had been done, returning it to the MSRF building during the tenth month or so.

At the conclusion of Phase 2, final documentation of the facility could be prepared and the training materials and final brochure generated. MSRF would be fully operational.

THIRD YEAR

The third year opens with MSRF fully operational to the extent that major studies of team performance, involving the simulation of systems on the scale of NTDS, can be supported. All of the major software and hardware components needed to perform research efforts of this magnitude should exist, with the exception of the Personality Skeleta discussed on page 137. If such an experiment is planned by this time, this year's MSRF development work should be devoted to the creation of the required Personality. If not, a study should be undertaken to determine the system (or systems) of greatest immediate interest to NPRDC, and effort should be expended to prepare these skeleta in advance to encourage large team studies and to reduce their elapsed time and expense.

COSTS

The cost estimates shown in Tables 9, 10, 11, and 12, while based on the best information available to us, are necessarily

open to revision. Public Works rates were furnished us by NPRDC. Prices for equipment to be purchased are list prices and do not include, for example, the effects of existing GSA or other applicable government procurement pricing. Firm quotations were not obtained from vendors because few are interested in guaranteeing prices for more than 90 days. Some are subject to negotiations; for example, the price of the primary display system might be reduced if further discussions with its vendor could identify a functionally equivalent configuration that would lead to reductions in the amount of engineering time required in its fabrication. Other items, such as acquisition of additional console modules in Phases 2 and 3, are somewhat arbitrary; while not supported by a detailed breakdown, they reflect the probable need of additional items (perhaps new devices developed since the time of this writing, or an increased stock to improve the facility's flexibility by that time).

The Appendix provides information concerning vendors for the recommended MSRF equipment.

TABLE 9
MSRF EXPENDITURES BY DEVELOPMENT PHASE

YEAR 1	
•Site preparation (Phase 1)	$ 65,000
•Observer station	7,000
•Basic shop facility (work surfaces/storage)	1,500
•Documentation	7,000
•One complete subject console	60,000
•Primary display system (3-console support)	250,000
•Voice communications system	5,400
•Tape drive and printer for supervisory computer	13,000
•Software development and system integration (including application I)	102,000
	$510,900
YEAR 2	
•Site preparation (Phase 2)	$ 85,000
•Separate program development station	10,000
•Minimal tools for shop	5,000
•Two additional subject consoles	90,000
•Stock of additional console components	10,000
•Software development and system integration (including application II)	90,000
	$290,000
YEAR 3	
•Final spares stocking	$ 20,000
•Software development and system integration (including application III)	100,000
•Stock of additional console components	10,000
	$130,000
	$930,900

TABLE 10
COST BREAKDOWN OF SUBJECT CONSOLES

ITEM		TOTAL
•Basic console rack and configuration kit		$ 1,000
•Console computer modules		11,000
•General and interprocessor interface design and parts	(first unit)	17,000
	(succeeding units)	4,000
•Power supplies		1,800
•Card cages for special hardware		200
•Internal cabling harness		300
•External cabling		300
•Assembly labor	(first unit)	12,000
	(succeeding units)	10,000
•High-resolution video monitor		2,000
CORE COST	(initial unit)	$45,600
	(succeeding units)	$30,600
•Trackball		$ 1,600
•10-key keyboard		100
•Full ASCII keyboard		400
•Reformattable plasma touch panels and interface	2 @ $5K	10,000
•Re-labelable analog meters	5 @ $40	200
•Pots	20 @ $10	200
•Switches and buttons	200 @ $ 4	800
•Force joystick		730
INEXPENSIVE MODULES		$14,030
TOTAL	(FIRST CONSOLE)	$59,630
TOTAL	(SUCCEEDING CONSOLES)	$44,630

AD-A044 234 HUMAN FACTORS RESEARCH INC GOLETA CALIF F/G 5/8

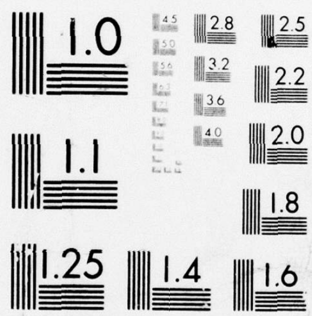
FUNCTIONAL REQUIREMENTS AND OTHER DESIGN FEATURES OF A MANNED S--ETC(U)

AUG 77 G V BAILEY, R T HENNESSY, C D WYLIE N00123-76-C-1245

UNCLASSIFIED NL

3 OF 3
AD
A044234

END
DATE
FILMED
10-77
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TABLE 11
PRICES OF SOME ADDITIONAL CONSOLE MODULES

512x512 8½"x8½" plasma panel	$4,000
LSI-11 interface for same	1,000
	$5,000
32x32 LED touch-panel for 8½"x8½" plasma panel	$1,000
LSI-11 interface for same	1,000
	$2,000
Trackballs	$1,300 - 1,700
Position joystick	$1,200
Alphanumeric-only video generators	$1,000
High-resoluti n video monitors with non-standard phosphors	$2,050

TABLE 12
DETAILED SITE PREPARATION COSTS

Power and grounding	$ 20,000
General refurbishment	70,000
Flooring	10,000
Air-conditioning	20,000
Additional partitioning for experimental area	10,000
Halon fire system	10,000
MINIMAL SITE PREP.	$140,000

APPENDIX
VENDORS OF RECOMMENDED MSRF EQUIPMENT

Console Structure

Dorlec Corporation
P.O. Box 182
Cherry Hill, New Jersey 08002

Digital Television Systems

Hazeltine Corporation
Greenlawn, New York 11740

Floor, Computer

Al-Lee Construction Company, Inc.
4609 West Jefferson
Los Angeles, California

Halon Fire Suppression System

J&M Carbonics
Los Angeles, California

Headset, Sound-Powered

David Clark Company, Inc.
360 Franklin Street
Worcester, Massachusetts 01604

Intercom

Talk-A-Phone
5013 North Kedzie Avenue
Chicago, Illinois 60625

Interface Modules

ADAC Corporation
15 Cummings Park
Woburn, Massachusetts 01801

Joystick

Measurement Systems, Inc.
523 West Avenue
Norwalk, Connecticut 06850

Mini/Microcomputers

Digital Equipment Corporation
Maynard, Massachusetts 01754

Monitor, High-Resolution

Conrac Division, Conrac Corporation
600 North Rimsdale Avenue
Covina, California 91722

Plasma Panels

Information Technology, Limited
706 Jackson Street
Monticello, Illinois 61856

Recorder, 4-Track

Sony, Model TC-277-4

Stiff Stick

Measurement Systems, Inc.
523 West Avenue
Norwalk, Connecticut 06850

Switches

Honeywell, Micro Switch Division
11 West Spring Street
Freeport, Illinois 61032

Switchcraft, Inc.
5555 North Elston
Chicago, Illinois 60630

Timecode Generator/Reader

Moxon, Inc.
2222 Michelson Drive
Irvine, California 92715

Trackball

Measurement Systems, Inc.
523 West Avenue
Norwalk, Connecticut 06850